

Enabling Radiative Transfer on AMR grids in CRASH

N. Hariharan^{1,4*}, L. Graziani^{1,5}, B. Ciardi¹, F. Miniati², H.-J. Bungartz³

¹*Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Strasse 1, 85748 Garching, Germany*

²*Department of Physics, ETH-Zürich, Wolfgang-Pauli-Strasse 27, CH-8093, Zürich, Switzerland*

³*Institut für Informatik, TU München, D-80290, München, Germany*

⁴*Intel Technology India Private Limited, #136 Old Airport Road, Bangalore-560008, India*

⁵*INAF Osservatorio Astronomico di Roma, Via Frascati 33, 00040, Monte Porzio Catone (RM), Italy*

Accepted 1988 December 15. Received 1988 December 14; in original form 1988 October 11

Key words: Cosmology:theory—Radiative Transfer—methods:numerical

ABSTRACT

We introduce **CRASH-AMR**, a new version of the cosmological Radiative Transfer (RT) code **CRASH**, enabled to use refined grids. This new feature allows us to attain higher resolution in our RT simulations and thus to describe more accurately ionisation and temperature patterns in high density regions. We have tested **CRASH-AMR** by simulating the evolution of an ionised region produced by a single source embedded in gas at constant density, as well as by a more realistic configuration of multiple sources in an inhomogeneous density field. While we find an excellent agreement with the previous version of **CRASH** when the AMR feature is disabled, showing that no numerical artifact has been introduced in **CRASH-AMR**, when additional refinement levels are used the code can simulate more accurately the physics of ionised gas in high density regions. This result has been attained at no computational loss, as RT simulations on AMR grids with maximum resolution equivalent to that of a uniform cartesian grid can be run with a gain of up to 60% in computational time.

1 INTRODUCTION

The observed universe shows a large variation in structures as we move along different scales. Independent observations of the distribution of faint radio sources, optically selected galaxies and the X-ray background show that our universe is homogeneous on a scale larger than 200 Mpc (see e.g. Wu et al. 1999 for a review), while at smaller scales it appears inhomogeneous due to the presence of a large number of structures: for example, galaxy clusters and galaxies at tens of Mpc, or stellar clusters and molecular clouds at parsec scales. In the Λ CDM concordance model of our universe, the presence of primordial density perturbations led to the gravitational collapse and cooling of gas in pre-existing dark-matter halos, and to the subsequent formation of radiating sources like stars and quasars (e.g. Mo et al. 2010). The chemical, mechanical and radiative feedback by these sources on their surroundings induces a complex interplay between the galaxy formation process and the evolution of the intergalactic medium (IGM; Ciardi & Ferrara 2005; Davé 2005; Barkana & Loeb 2007; Meiksin 2009). Among these feedback effects, a prominent place is occupied by the IGM reionisation process, which denotes the transition from a neutral intergalactic gas to an IGM which is (almost) fully ionised in its hydrogen component by $z \sim 6$ (e.g. Fan et al.

2006), while helium reionisation is believed to be complete at $z \sim 2.7$ (e.g. Madau & Meiksin 1994; Compostella et al. 2013).

A large number of simulations have been done to understand the formation of the first structures and the subsequent galaxy formation and evolution. Examples of collisionless dark-matter simulations include Shaw et al. (2006), Boylan-Kolchin et al. (2009), Kim et al. (2011), Angulo et al. (2012) and Harnois-Déraps et al. (2013), while e.g. Springel (2005), Vogelsberger et al. (2014) and Schaye et al. (2014) follow also the gas dynamics. To study radiative feedback, radiative transfer (RT) simulations are usually used as post-processing tools for hydro codes or N-body codes, and they form a major aspect of the study of structure formation in general, and the IGM reionisation in particular (e.g. Ciardi et al. 2003; Iliev et al. 2006a, 2007; Zahn et al. 2007; Trac & Cen 2008; Ahn et al. 2012; Iliev et al. 2014; Graziani et al. 2015). Simulations where the gas dynamics and RT effects are self-consistently accounted for have also been performed (e.g. Gnedin & Ostriker 1997; Semelin et al. 2007; Gnedin 2014; Gnedin & Kaurov 2014; Pawlik et al. 2015). Large scale hydro and RT simulations require a large amount of computational and memory resources to handle spatial resolutions spanning several orders of magnitude. In the case of grid-based codes, this problem has been addressed to a great

extent by the use of Adaptive Mesh Refinement (AMR) schemes (e.g. Kravtsov et al. 1997; Teyssier 2002; Miniati & Colella 2007; Bryan et al. 2014), which provide a great amount of theoretical and algorithmic resources, and have been used extensively to solve a wide range of problems, such as numerical relativity, global weather and nuclear fusion modelling (e.g. Plewa et al. 2003).

AMR schemes use a mesh or a grid to describe the physical domain and to progressively increase the grid resolution in confined regions of the mesh, based on a set of refinement criteria. By selectively increasing the resolution only in the interesting part of the domain, AMR methods optimise the global memory and computational resource requirements. The original idea of AMR was to introduce a finer grid in regions of higher numerical error, which can be identified by density gradients or by Richardson extrapolation (Berger & Oliger 1984; Berger & Colella 1989). However, AMR can be employed very flexibly, and different refinement criteria can be chosen. For example, in cosmological applications it is customary to adopt a Lagrangian criterion in which cells are refined when their mass is above a given threshold (e.g. Wise & Abel 2011). Alternatively, it is also possible to refine specific volumes in a Eulerian style, which is advantageous for example if one is interested in resolving turbulent motions inside cosmological halos (e.g. Miniati 2014).

AMR is also very flexible from an algorithmic point of view. For example, it naturally supports divergence-free magnetohydrodynamics (e.g. Miniati & Martin 2011; Teyssier et al. 2006; Lee & Deane 2009), it has been used to make detailed studies of thermonuclear flashes in FLASH (Fryxell et al. 2000), and also to simulate the formation of large-scale cosmological structures in ENZO (see e.g. Bryan et al. 2014 and Plewa et al. 2003 for more examples).

Many hydro codes make use of AMR schemes to carry out gas dynamic simulations over a range of spatial scales. Some of them are also coupled with RT schemes to perform self-consistent simulations where the RT feedback is accounted for in the dynamical evolution of the gas. The Hydrodynamics Adaptive Refinement Tree (HART) code (Gnedin et al. 2009) uses the OTVET approximation for the 3D RT implementation (Gnedin & Abel 2001), while the cosmological hydrodynamics code RAMSES (Rosdahl et al. 2013), designed for simulations of structure formation, incorporates RT using the M1 closure formalism (Levermore 1984). Both HART and RAMSES make use of the Fully Threaded Tree described in Khokhlov (1998) to implement AMR. For the RT ENZO uses an adaptive ray-tracing scheme implemented in the HEALPix library (Abel & Wandelt 2002; Górski et al. 2005). The interested reader can find more examples in Iliev et al. (2009).

Enabling AMR in a stand-alone RT code makes it suitable for post-processing the output of many grid-based hydro codes that use the same AMR logic. By representing the regions of interest with high resolution grids, we can follow the details of the growth of the ionised bubbles around the sources and understand the impact of the RT feedback effects on the surrounding environment. A number of stand-alone RT codes implementing AMR exist: some examples are RADAMESH (Cantalupo & Porciani 2011), which is a Monte Carlo (MC) RT code with a ray-tracing scheme, and FTTE (Razoumov & Cardall 2005), which implements a scheme to perform RT on refined grids in the presence of diffuse and

point sources. A final example is IFT (Alvarez et al. 2006), which has been developed to explicitly follow the I-front around a point source.

From the above examples, it is clear that AMR-enabled stand-alone RT codes are increasingly in use and they stand to benefit from the advantages that AMR provides. Keeping this in mind, we have developed CRASH-AMR, a novel implementation of the RT code CRASH (Ciardi et al. 2001; Maselli et al. 2003; Maselli & Ferrara 2005; Maselli et al. 2009; Pierleoni et al. 2009; Partl et al. 2011; Graziani et al. 2013), which is interfaced with the open source AMR library CHOMBO (Adams et al. 2011) to perform RT simulations on AMR grids. The aim of this paper is to introduce this new version, to discuss its implementation details and to show the tests that we have done to verify and validate CRASH-AMR.

The paper is structured as follows. In Section 2 we briefly introduce AMR and its different schemes, the CHOMBO library is also presented in this context and we mention some of its applications. In Section 3 we introduce the ray tracing implementation in the CRASH code. In Section 4 we discuss the method used to couple CRASH and CHOMBO to enable AMR in CRASH. The tests performed to verify our code are discussed in Section 5 and the results are summarised in Section 6. The advantages of the code in terms of computational costs is highlighted in Appendix A and B. Here, we discuss the tests done to show the dependence on grid-resolution and performance of the code. We discuss code performance in terms of run-time and correctness of results when compared to running RT simulations on uniform grids.

2 BASICS OF AMR AND THE CHOMBO LIBRARY

AMR is a successful technique when a problem presents a highly inhomogeneous spatial distribution implying that some regions require additional resolution, i.e. additional refinement. An AMR mesh is defined as structured (SAMR) if its cells (typically cartesian) are connected in a regular geometry. Un-structured grids composed of triangular or tetrahedral cells without a regular connectivity can also be defined (see for example Mavriplis 1997; Khokhlov 1998; Springel 2011 and Paardekooper et al. 2008). We limit our discussion to SAMR schemes in this paper.

There are many alternative ways to refine a certain sub-domain using different SAMR schemes. In the *Cell-based* (CBAMR) scheme each cell of the grid is refined as and when required and generally a quad-tree (2D) or and oct-tree (3D) forms the hierarchical structure relating the “coarser” (parent) and “finer” (child) cells (Young et al. 1991). The *Block-structured* AMR (BSAMR) scheme tags and refines a region of the grid, by some integer factor, based on pre-assigned criteria (Berger & Oliger 1984). The various refinements are typically organised in a hierarchical structure connecting coarser layers (parent level) with refined layers (child level); the grids are then stored and maintained independent from each other (Berger & Oliger 1984). When the cells that need to be refined are clustered together to form disjoint rectangular patches, BSAMR schemes are called *Patch-based* (PBAMR; Dai 2010). Both BSAMR and PBAMR involve refining a specific region in the grid rather than a single

cell, and so tend to be used synonymously in AMR-related literature.

Hereafter, we will limit our discussion to the PBAMR scheme and look at the details involved for its usage in our RT code CRASH.

To enable CRASH to process AMR-based grids we have adopted the open-source AMR library CHOMBO (Adams et al. 2011). CHOMBO¹ is actively developed at Lawrence Berkeley National Laboratory and implements a PBAMR scheme on a C++ framework to solve systems of hyperbolic, parabolic and elliptic partial differential equations. It has been successfully used by many gas dynamic codes such as CHARM (Miniati & Colella 2007), FLASH (Dubey et al. 2014) and PLUTO (Mignone et al. 2012), and hence is suitable for our purposes as well. One could, in principle, also post-process the outputs of codes adopting similar BSAMR framework as done, for example, by ENZO (Bryan et al. 2014). This would, however, require an intermediate step to convert both the grid hierarchy and file formats in a data format suitable for CHOMBO. The development of this interface will be taken up in the future; in this paper we limit to the data that has been directly produced using the CHOMBO library or the codes adopting it, for example CHARM. Test 2, in Section 5.2, shows one example of CHARM output post-processed by CRASH-AMR².

The library is organised into a hierarchy of classes, each of which provides a specific functionality for incorporating AMR into a stand-alone code with minimum effort, so that the software developers only need to focus on implementing the physics. Some necessary but routine tasks, associated for example with grid generation, management, refinement and time-stepping, are automatically managed by the library. Hereafter, we describe some of the functionalities implemented by CHOMBO that have been extensively used in CRASH-AMR. For each of them, the adopted C++ class is also indicated.

• PBAMR organisation in a hierarchy of levels.

The PBAMR-AMR scheme implemented by the library consists of various refinement levels L organized in a hierarchy starting from the base level ($L = 0$), and extending up to a final level with index $L = N - 1$, where N is the total number of levels in the hierarchy. Each level has its own resolution defined by $r_0 \cdot r^N$, where r_0 is the resolution of level $L = 0$ and r is the “refinement ratio”, i.e. the ratio of the resolution between two contiguous levels. While the entire AMR scheme is represented by a C++ class *AMR*, each refinement level L in the AMR hierarchy is implemented in CHOMBO by the class *AMRLevel*. Each *AMRLevel* class implements pointers to its relevant parent in the hierarchy (i.e. level $L - 1$) and its child (i.e. $L + 1$) to allow an easy traversal of the entire hierarchy of refinement levels.

¹ Chombo is a Swahili word meaning “tool” or “container”.

² The adoption of output data produced by codes implementing different AMR schemes (e.g. RAMSES, which uses a cell-based scheme (Khokhlov 1998)) could require a non-negligible effort in converting both the grid representations and the file formats. The ray-tracing scheme, described in Section 4, should also sensitively adapt to the different nesting geometry of the AMR levels. The adoption of different AMR schemes is then beyond the scope of the following paper.

• Representation of each level as composition of cell boxes.

In the PBAMR scheme of CHOMBO each level is composed by cells which should be seen as minimum units of space with assigned width. Cells are organized in rectangular grids called “boxes” each of which occupies a unique location in the level and it is disjoint from the others, i.e. a cell at a particular refinement level can belong to only one box. CHOMBO represents these boxes by the *Box* class. Each box occupies a unique location in the 3D space, and its coordinates are provided by the *smallEnd* and *bigEnd* functions of the *Box* class. A refinement level is represented as an array of disjoint boxes (*DisjointBoxLayout*) which can be traversed with iterator classes (e.g., the *DataIterator* class) to access a single box.

Boxes that lie adjacent to a box at the same refinement level are said to be its neighbors and can be accessed through the *NeighborIterator* class. Subset of boxes can also be grouped together in an array called “Fortran Array Box” (corresponding to the class *FArrayBox*) to allow an easy and fast access to the subset data for retrieval or update operations.

• Interactions between different refinement levels.

When a coarsening or refinement is requested at some level L , it operates on certain box(es) with a given refinement ratio r , also implying that the box(es) can have multiple child boxes (i.e. at level $L + 1$) and can lie over multiple parent boxes (i.e. at level $L - 1$). The boxes at level $L - 1$ or $L + 1$ whose intersection with a box at level L is non-empty are its parent and child boxes, respectively.

When physical quantities representing continuous fields are refined in space (i.e. the result is stored across different levels) the continuity of their gradients must be ensured. For this reason the AMR scheme of CHOMBO adopts interpolation and averaging methods at the interface of grids. A typical example of these smoothing operations occur when the data on the grids are loaded to set up the initial conditions (ICs) of a hydro simulation. To initialise the finer grids from the existing coarse grids, the *FineInterp* class is used, while to update the coarse grids with the data on the finer grids the *CoarseAverage* class is adopted.

• **Data storage and grid I/O.** CRASH-AMR adopts the HDF5 data format standard³ to store the RT results, so that they can be easily post-processed and visualised by using state-of-the-art visualisation software like Visit⁴ or Paraview⁵.

3 RADIATIVE TRANSFER CODE CRASH

CRASH is a 3D MC RT code that can self-consistently follow the formation and evolution of ionised regions created by sources present in a static and inhomogeneous gas environment; the gas consists of H, He and metals. The temperature evolution of the gas is calculated self-consistently. Additionally, the code can account for an arbitrary number of point sources, as well as a UV background.

Our work is based on CRASH version 3 (CRASH3; refer

³ <http://www.hdfgroup.org/>

⁴ <https://wci.llnl.gov/codes/visit/home.html>

⁵ <http://www.paraview.org/>

to [Graziani et al. 2013](#) and references therein for more details), and the developments presented here contribute to **CRASH-AMR** by using **CRASH3** as baseline but, for simplicity, without the inclusion of metals. In this section, we discuss the **CRASH** code briefly, and we provide only the details relevant to the AMR implementation.

CRASH works by assigning the ICs onto a static, regular 3D grid which specifies the gas number density n_{gas} , temperature T , the H, He ionisation fractions ($x_{\text{HII}}, x_{\text{HeII}}, x_{\text{HeIII}}$), the source coordinates, luminosity L and spectral energy distribution (SED) S . The radiation from each source is discretised into photon-packets represented by N_ν frequency bins, each containing $N_{p,\nu}$ photons as determined by the SED, which are propagated along the rays casted in random directions from the point sources. The simulation proceeds by emitting photon-packets from all the sources and propagating them along the rays until the end of the simulation time.

We discuss briefly the ray tracing routine of **CRASH** in the following paragraphs. Consider a ray along which a packet propagates by crossing a series of cells. For each cell l that is crossed, **CRASH** calculates the casted path δ_l and the corresponding optical depth of the cell as

$$\begin{aligned} \tau &= \tau_{\text{HI}} + \tau_{\text{HeI}} + \tau_{\text{HeII}} \\ &= [\sigma_{\text{HI}}(\nu)n_{\text{HI}} + \sigma_{\text{HeI}}(\nu)n_{\text{HeI}} + \sigma_{\text{HeII}}(\nu)n_{\text{HeII}}]\delta_l, \end{aligned} \quad (1)$$

where n_A and σ_A are the number density and cross section of the absorber $A = \text{H I}, \text{He I}, \text{He II}$. If the packet reaches the cell with a photon content of N_γ , then the number of photons absorbed in the cell is given by

$$N_\gamma^l = N_\gamma(1 - e^{-\tau}). \quad (2)$$

N_γ^l is then used to calculate the ionisation, recombination fractions and temperature equations that regulate the physical state of the gas (Section 2.4 of [Maselli et al. 2003](#)). The angular direction of the ray and coordinates of the current cell are used to calculate the coordinates of the next cell that the ray will cross; this is repeated until the photon content in the packet is extinguished or, if periodic boundary conditions are not applied, the packet exits the grid. We finally note that since we use **CRASH-AMR** in a post-processing mode, operations of data smoothing, described in the section above, are confined to the initialisation of the RT and do not impact the ray tracing algorithm.

4 COUPLING CRASH AND CHOMBO

In this section we describe how **CRASH** and **CHOMBO** are coupled to implement **CRASH-AMR**; we discuss both the adopted methodology and the solutions we found to the various technical issues that occurred during the code development.

CRASH-AMR is implemented in Fortran 2003, while **CHOMBO** is a C++ code. We have created a C interface between Fortran and C++ to allow **CHOMBO** to communicate and share information with **CRASH-AMR** by using the interoperability features implemented in the programming languages specifications. As discussed in Section 2, **CRASH-AMR** uses the grid representation of **CHOMBO** (i.e. the *Box* class

and its subclasses) to store the physical variables that have a spatial representation, e.g. $n_{\text{gas}}, x_{\text{HII}}, x_{\text{HeII}}, x_{\text{HeIII}}$ and T .

In the ray tracing algorithm implemented in **CRASH-AMR** (see Sec. 3 for more details), the interaction of radiation with matter is computed in each crossed cell by solving the ionisation and temperature equations. This implies that when a photon packet propagates through the domain it can cross many refined regions containing multiple PBAMR **CHOMBO** patches. Moving through different AMR layers implies that the instances of the *Box* classes representing collection of cells at each refinement level are continuously accessed during the travel of each photon packet (see Sec. 3) to update and store the physical quantities.

The computational cost of a continuous and inefficient access to the **CHOMBO** library could impact the global RT performance. In fact, depending on the chosen resolution in space and the maximum refinement level provided by the gas dynamics simulation, a single ray could traverse a large number of cells spanning different refinement levels. This makes the box iteration computationally inefficient when repeated for the large number of rays required by the MC convergence (typically larger than 10^7). Note that this is not the way PBAMR libraries are normally used in hydro codes to access the information: the patch-based scheme implemented in **CHOMBO** is in fact very efficient in the management of memory and parallel computational resources, but provides information at the grid level instead of at the cell-based quantities, as required by the **CRASH-AMR** RT scheme. As further complication, a realistic RT simulation generally involves an irregular distribution in space of the sources from which a large number of rays is emitted in random directions, implying that the boxes at each refinement level are not accessed contiguously. Consequently, the standard interface provided by the **CHOMBO** library cannot be simply re-used in **CRASH-AMR**. To resolve this issue we have developed a novel Fortran data structure in **CRASH-AMR**, minimising the run-time overhead to access and iterate the AMR layers, as described below.

During each ray traversal the photon-packet information has to be propagated through the AMR grid hierarchy. If we assume that the source lies in the highest refinement level, then the ray will propagate, starting from that level, to the coarser levels below and move back to finer levels if present. At each step of the RT simulation, the data structure needs to know the refinement level a cell belongs to, whether the cell is refined or not, and which box contains the refined cell; the way all these quantities are accessed and the mapping between **CRASH** and **CHOMBO** data is described in the following paragraphs. Hereafter, we will refer to the grid with the lowest resolution as ‘base grid’, while the refined grids will be referred to as ‘refined levels’.

Figure 1 shows the data representation in **CHOMBO** on the left-hand side (see also Section 2 for more details), and the **CRASH** equivalent on the right-hand. We also use similar colours in both sides to represent corresponding boxes at a given refinement level. A simplified but representative **CHOMBO** hierarchy, consisting of the base level 0 and its refinements from 1 to $L - 1$, is shown in the picture; for clarity purposes we just represent the boxes at levels 0, 1 and 2. To help the reader in connecting this picture with the abstract data representation provided in Section 3, we point out that each refinement level (dashed boxes on the left)

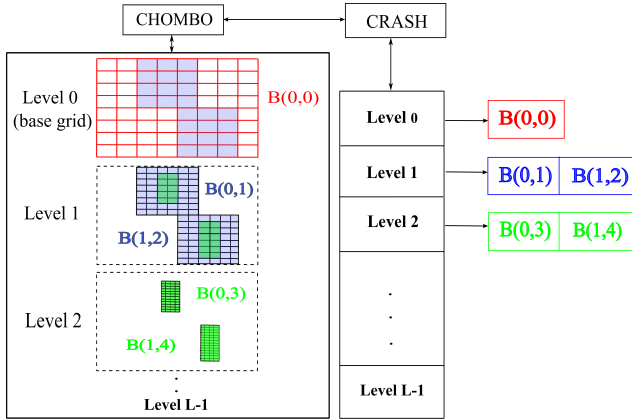


Figure 1. Interface between CRASH and CHOMBO, the grid hierarchy in CHOMBO being reflected through the data structure built in CRASH. The AMR grids, stored as an array of boxes in CHOMBO, are stored in CRASH on a level basis, with pointers to the grid data at each level. Each box has an associated *localID* and *globalID*. See text for more details.

is implemented in computer memory by an instance of the *AMRLevel* class, while the array of boxes at each level is implemented by the *DisjointBoxLayout* class. Each box, for example the $B(0,0)$ at base level 0 (red box on the left), is an instance of the *Box* class.

The CRASH counterpart of the AMR hierarchy is mapped on the right-hand side of Figure 1: the base grid is represented as refinement level 0, while the L AMR grid levels are mapped with an array containing pointers to specific properties of each box. The boxes at each level are uniquely identified, on both sides, by an associated *localID* and *globalID*. We first use the *localID* to get direct access to the right box in the *DisjointBoxLayout* array, and then to the physical variables of interest during the RT simulation. The corresponding *globalID* along with the refinement level provides an index into one of the arrays on the CRASH side. For example, box $B(0,1)$ at refinement Level 1 has a *localID* of 0 and a *globalID* of 1. The value of the *localID* indicates that $B(0,1)$ is at position 0 of the disjoint box array storing all the boxes at that level, and the physical data can be finally accessed through its *FArrayBox* (see Sec. 2 for more details on the classes). The start and end coordinates of the box, which determine its size, are also stored to allow the ray tracing algorithm to recognise if a ray has exited a box at a given refinement level. Along with the physical data, we store in each cell the *globalID* of the box it belongs to. For cells that are covered by the cells of a refined box, we store the *globalID* of the refined box. This is used to determine if a cell that the ray is passing through is refined or not. Additionally, since a PBAMR scheme allows a refined box to lie over multiple coarse boxes, we keep a list of all parents. This is done by storing, for each box, the *globalIDs* of its parent(s). Finally, we also store a neighbor list containing the *globalIDs* of all boxes that are neighbor(s) to a box.

The following paragraphs describe how the new data structure is used during the ray tracing algorithm. First note that the sources emitting photons do not move across the grid during a RT simulation. Hence, given the refinement level the source lies in and its coordinates we need to look for the *globalID* of the box, containing the source, only once.

This can then be used to index into the arrays in CRASH, to get the right box and the physical data stored at the source coordinates. Then, during the propagation of the photon packet, at each cell crossing, the following scenarios apply:

- (a) the ray might escape the grid and then we no longer follow it unless periodic boundary conditions are applied;
- (b) the photon content of the packet is completely absorbed and then the propagation stops;
- (c) the ray crosses the cell and enters a new cell at the same AMR level;
- (d) the ray crosses the cell and enters another cell at a finer (or coarser) AMR level.

While cases (a) and (b) do not need further comments, for case (c) the new cell might lie in the same box or it might enter a new one. In the former case, we just continue the ray propagation as described earlier; in the latter case, we use the coordinates of the new cell the ray is in and search the neighbors of the box that the ray was previously in, at the same refinement level, for the box that contains the new cell. Finally, case (d) needs a different approach because the cell optical depth, calculated using the casted path, depends on the refinement level the ray is crossing through (see Eq. 1). Here, again, different scenarios apply:

- (1) the ray enters a finer level. Given the new cell coordinates, we check if the *globalID* stored in the cell is the same as the *globalID* of the box the ray was in. If not, this indicates that the cell is covered by a refined cell. The *globalID* stored in the new cell is then used to find the refined box and the corresponding cell. Finally the ray is moved to the finer level;
- (2) the ray enters a coarser level. We search the parent list, of the box the ray was previously in, for the new box containing the new cell.

Once the new box is found in the neighbor list for case (c) or in the parent list for (2) we recursively move the ray, as in case (1), to a finer level if the neighbor or parent is refined. The same procedure is repeated until cases (a) and/or (b) apply, and the photon packet propagation stops.

Figure 2 shows the logical workflow described above. In this diagram the solid lines of the flowchart indicate the standard steps of the CRASH ray tracing algorithm, while the dashed lines indicate the steps performed by the CRASH-to-CHOMBO interface to move across the various CHOMBO boxes (CHBox) of the AMR side. The many scenarios described above can be visually followed by colours, as shown in the legend of the figure.

It is important to note that, although we use CHOMBO to initialise and store the AMR grid, once this data has been mapped onto the CRASH side, our implementation does not call any CHOMBO routines during the RT simulation. The data structure is used purely to cross the levels and have a fast access to the grid data. As a consequence of this architectural choice, there is no overhead of using CHOMBO during the ray-tracing routine but the time needed to find the right cell.

As a final consideration, we want to emphasise that the above features are included as a separate functionality, allowing the user to enable or disable the use of AMR grids to do RT simulations and use the traditional data storage setting up the simulation ICs only on the base grid. If the AMR

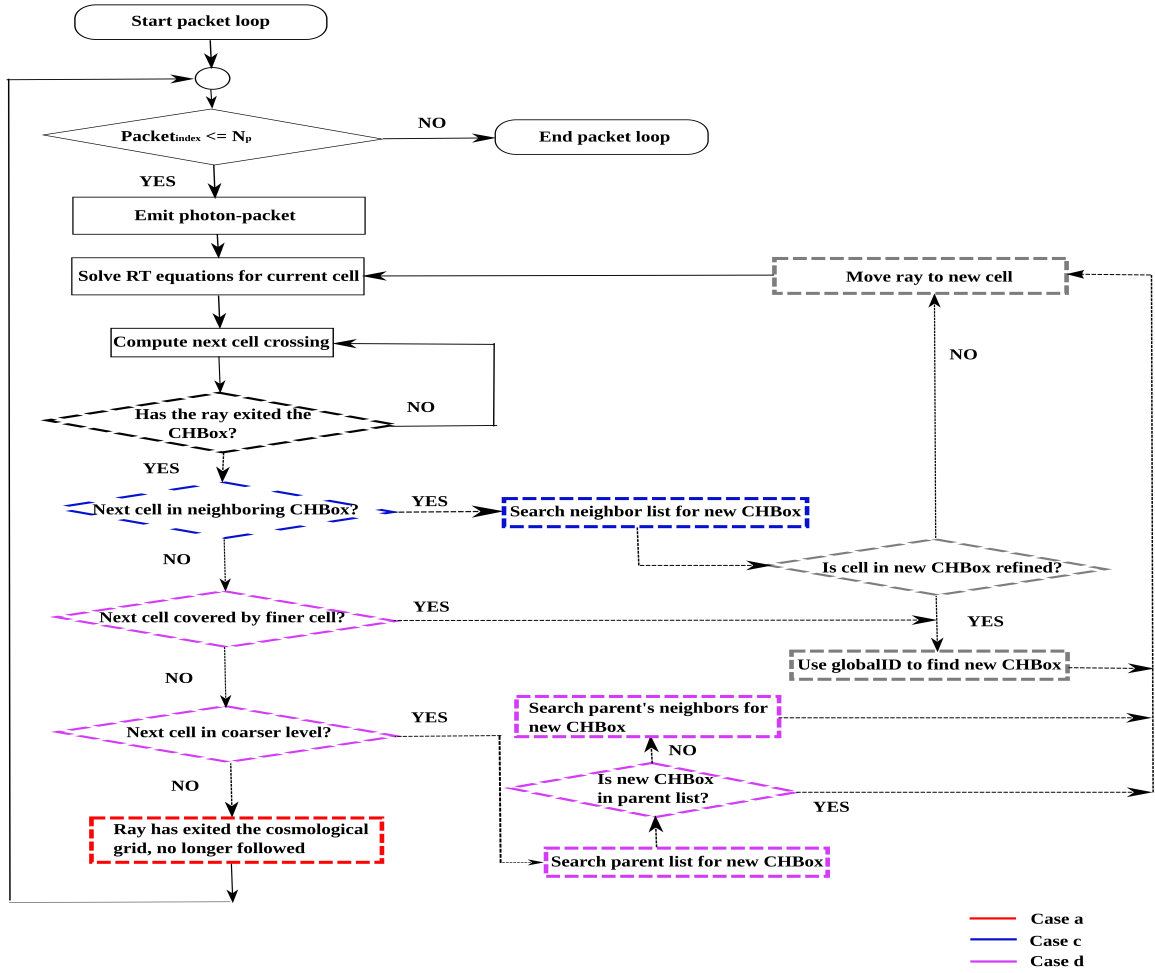


Figure 2. CRASH-AMR flowchart describing the interplay between ray tracing and the CRASH-to-CHOMBO interface when a photon packet travels through many AMR levels. The boxes with solid lines indicate the CRASH-side of the algorithm, while those with dashed lines indicate the CHOMBO interface-side. The solid/dashed lines connecting the boxes indicate if the CRASH/CHOMBO interface-side of the algorithm, respectively, is being called. Various colours refer to the different algorithmic scenarios that could happen during a photon packet propagation. See text for more details.

functionality is enabled, then the user can either run simple tests with pre-defined refinement criteria, or more realistic cases with AMR grids provided by hydro codes, as shown in the following section. By adopting pre-defined refinement criteria, the user could decide, for example, to refine an arbitrary part of the base grid and set up specific test cases, while for realistic gas configurations, the refinement is generally determined by the hydro code, and CRASH-AMR operates in post-processing mode.

5 TESTS AND RESULTS

In this section we show the results of some of the tests we have performed to guarantee the reliability of the new AMR implementation. We run a number of test cases in idealised configurations; in Test 1 we compare CRASH-AMR with the AMR functionality disabled to CRASH3, and then we compare results with/without the AMR functionality enabled. These set-ups are useful to check the numerical noise introduced by the presence of the AMR grids on the RT algorithm.

In Test 2, we apply CRASH-AMR to a realistic density field from the CHARM simulations described in Miniati & Colella (2007). Additionally, in Appendix B we take a further look at the performance of the code in terms of run times and correctness of results. Henceforth, we use d to represent the comoving distance from a point source, the units in kpc or Mpc are indicated accordingly.

5.1 Test 1: Strömgren sphere in a H+He medium

We have set up a test equivalent to Test 2 of the Radiative Transfer Code Comparison Project (RTCCP; Iliev et al. 2006b). The test simulates the evolution of an ionised region around a single point source located at the grid origin (1,1,1) in a box of side length $L = 6.6$ kpc and mapped on a grid of 128^3 cells. The source is assumed to be steady with an ionising rate of $\dot{N}_\gamma = 5 \cdot 10^{48} \text{ photons} \cdot \text{s}^{-1}$ and an associated black-body spectrum at temperature $T_{BB} = 10^5$ K. The volume is filled by a uniform and static gas of number density $n_{\text{gas}} = 10^{-3} \text{ cm}^{-3}$, containing H (92% by number) and He (8%). The gas is assumed to be fully neutral and

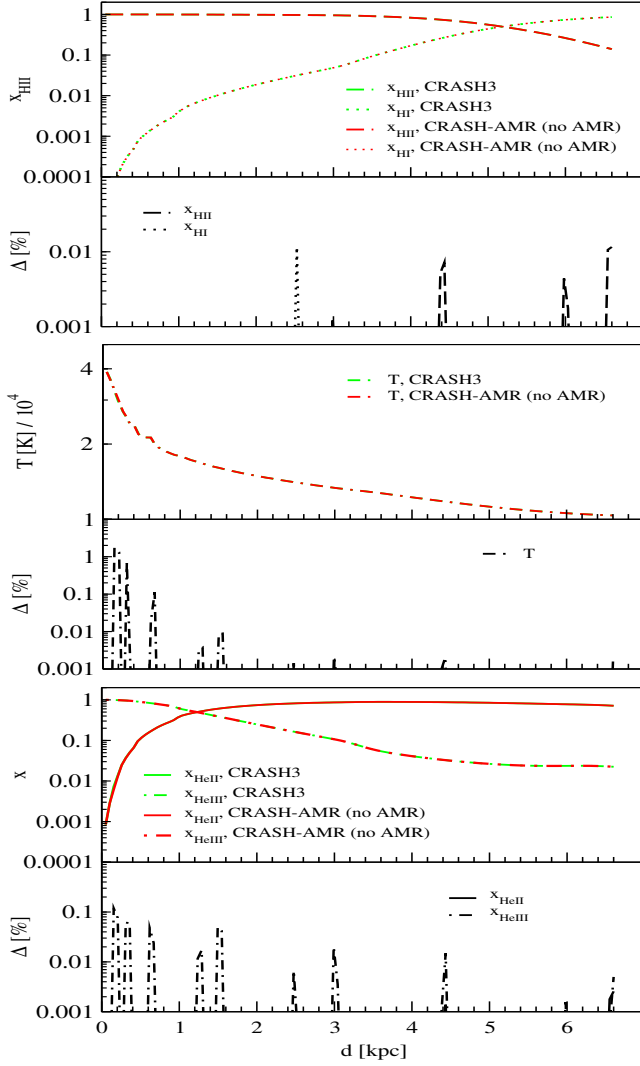


Figure 3. Spherically-averaged profiles at time $t = 500$ Myr for Test 1a. The colours refer to CRASH3 (green) and CRASH-AMR (AMR disabled; red). The bottom sub-panels show Δ between the CRASH3 and CRASH-AMR (AMR disabled) results. **Top:** Profiles of x_{HII} (dashed lines) and x_{HI} (dotted lines). **Middle:** Profile of T (dash-dot lines). **Bottom:** Profiles of x_{HeII} (solid lines) and x_{HeIII} (dot-dash lines).

at a temperature $T=100$ K, which is then calculated self-consistently with the progress of ionisation for a simulation time $t_{\text{sim}} = 500$ Myr, starting at redshift $z = 0.1$. We output the results at intermediate times $t = 10, 50, 100$ and 200 Myr as in the original set-up. It should be noted that simpler tests (e.g., with a gas composed by H only or with the temperature kept constant) have been run as well, and give results similar to those discussed in the following.

5.1.1 Test 1a: AMR disabled

To verify that the changes done to enable RT on AMR grids do not introduce any numerical noise, we have run Test 1 with AMR disabled in CRASH-AMR and compared the results to those from CRASH3.

The outcome is given in Figure 3, where each panel

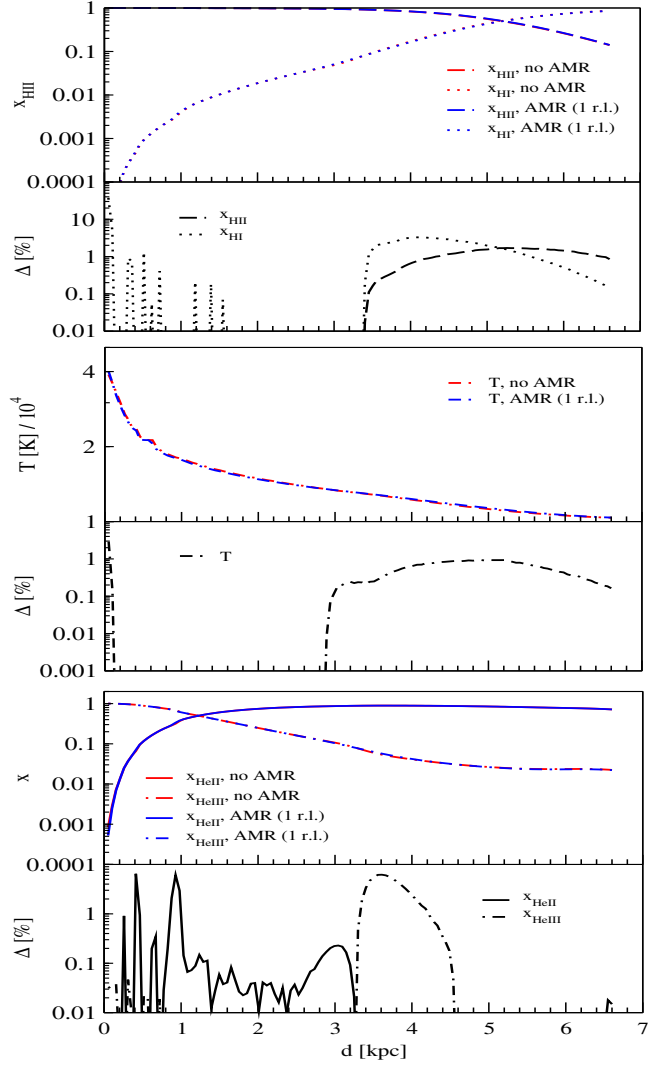


Figure 4. Spherically-averaged profiles at time $t = 500$ Myr for Test 1b. The colours refer to CRASH-AMR (AMR disabled; red) and CRASH-AMR with one refinement level (1 r.l.; blue). The bottom sub-panels show Δ between the CRASH-AMR (AMR disabled) and CRASH-AMR (1 r.l.) results. **Top:** Profiles of x_{HII} (dashed lines) and x_{HI} (dotted lines). **Middle:** Profile of T (dash-dot lines). **Bottom:** Profiles of x_{HeII} (solid lines) and x_{HeIII} (dot-dash lines).

shows spherically-averaged physical quantities as a function of d , together with the percentage difference (Δ) between the CRASH3 and the CRASH-AMR results. We define $\Delta = (R_{\text{ref}} - R_i) \cdot 100 / R_{\text{ref}}$, where R_{ref} and R_i refer to results of CRASH3 and CRASH-AMR, respectively. From the Figure it is clear that there is no significant difference between the two codes, while the spikes that we see in Δ (with a maximum of $\sim 1\%$, but mostly below 0.1%) are due to numerical artefacts caused by optimisation of CRASH-AMR involving rearranging of double-precision floating point arithmetic expressions, and are not due to the changes associated with CHOMBO. This shows that the AMR feature in CRASH-AMR is isolated from the rest of the code and can be disabled without introducing any numerical noise into the results.

5.1.2 Test 1b: AMR enabled

Our next step has been to run the previous test with AMR enabled, refining 100^3 cells around the source by a factor of 2, with the ICs set up on the base grid as well as on the refined grid. The source is now surrounded by a cartesian grid, at an equivalent resolution of 256^3 , of ~ 5 kpc. The results are given in Figure 4, where each panel shows spherically-averaged physical quantities as a function of d , together with the Δ between the results from CRASH-AMR (AMR disabled, R_{ref}) and CRASH-AMR with one refinement level (1 r.l., R_i). Here we find that Δ can be as high as 10% for x_{HI} and x_{HeIII} cells very close to the source ($d \sim 0 - 1$ kpc), and at a distance of 3 - 6 kpc, in the partially ionised region. More typical values, though, do not exceed 1% for all the physical quantities.

From Section 5.1.1 we know that, with the exception of the spikes observed for x_{HI} , the differences seen are not due to the implementation in CRASH-AMR, but they must rather be associated to the higher grid resolution in the refined region. Also, these differences are not due to any spurious noise produced by this particular method of calculating the average, but is rather due to the higher resolution being used. When we have a grid at a higher refinement level all the refined cells that cover a coarse cell might not lie within a given radius from the source and thus not contribute to the spherical average. This effect is more prominent in the partially ionised regions where two adjacent cells might not have the same e.g. x_{HI} values, unlike a fully ionised region where x_{HI} is 1. If a cell does not lie within the given radius, the average value will differ. Also note that Helium has a recombination rate five times higher than that of H. As a result, the helium components show more sensitivity and we get differences of $\sim 1 - 10\%$ in many cells. CRASH-AMR will then provide a better description of the regions with fully ionised helium, generally more confined to the brightest and x-rays luminous sources (e.g. quasars).

Additional tests showing the dependence of the results on the grid resolution are detailed in Appendix A.

5.2 Test 2: a realistic density field

In this section we apply CRASH-AMR on a density field snapshot obtained from a simulation run within the Santa Barbara Cluster Comparison Project, where the formation of a galaxy cluster in a standard CDM universe is followed (Frenk et al. 1999). The simulation has been performed by the hydro code CHARM (Miniati & Colella 2007) in a box size $L = 64$ Mpc (comoving) at redshift $z = 0.1$. The cosmological parameters are $\Omega_m = 1$, $\Omega_b = 0.1$, $\Omega_l = 0$ and $H_0 = 50$ km s $^{-1}$ Mpc $^{-1}$. The simulation is initialised at $z = 40$ with a base grid of 64^3 cells representing a box of 64^3 Mpc 3 comoving, and a grid of 128^3 cells placed at the centre of the base grid and representing a box of comoving length 32 Mpc. Only the central region is refined based on a local density criterion, with a refinement ratio of 2. At the end of the simulation there are six refinement levels in total, along with the base grid. The cell width at the coarsest level is 1 Mpc, while that at the finest level, with an equivalent resolution of 4096^3 cells, is 15 kpc. The code CHARM adopts the CHOMBO library to implement the AMR functionality, and the HDF5 files available from the output

of this simulation can be immediately used as an input to CRASH-AMR by extracting the necessary information from the HDF5 metadata. As the simulation does not provide information on the star formation, we define the point source locations associating them to the gas density peaks at the most refined level.

We set up the following RT simulations:

- (a) multiple point sources placed at the highest refinement level, at locations far enough so that when moved to lower refinement levels they do not gather. The sources are monochromatic, the gas temperature is kept constant throughout the simulation;
- (b) as (a), but now the point sources are placed at locations close enough so that they can be gathered at the lower refinement levels;
- (c) same point source locations as (a), but with a black-body spectrum, and the gas temperature is calculated self-consistently with the progress of ionisation during the simulation.

In all cases the point sources are located within high density peaks, chosen to ensure that the criteria mentioned above are satisfied for our tests. Since the base grid is refined only in the central region within which the higher refinement levels also lie, the resolution at large distances from the point sources is the same for all cases.

We set a reference ionisation rate, $\dot{N}_{\gamma, \text{ref}}$, for the source in the highest gas density peak. For the other sources i :

$$\dot{N}_{\gamma, i} = \frac{\dot{N}_{\gamma, \text{ref}} \cdot m_i}{m_{\text{ref}}}, \quad (3)$$

where m_{ref} and m_i are the mass in the cell containing the reference source and source i , respectively. The initial temperature is $T = 100$ K and the gas is assumed to be fully neutral. The simulation time is 500 Myr.

To emphasise the advantage of an AMR scheme, we compare results of simulations run with different refinement levels. Additionally, as mentioned above, the sources are located at the highest refinement level, so if one or more of them lie within the same cell at the coarser levels, we consider them to be a single source with luminosity given by the sum of the corresponding luminosities at the finest level.

To ensure a good convergence of the MC code, we sample the radiation field with a number of photon packets high enough to reach convergence for each test case run at different refinement levels. We find that the MC scheme converges with 10^8 photon packets per source (0.07% difference in volume averaged x_{HI} values between two test cases with 10^8 and 10^9 photon packets per source). However, the convergence of the MC scheme is very much problem dependent, hence we do not discuss this further.

5.2.1 Test 2a: multiple sources set far apart - constant T

Here we place twenty point sources far enough from each other to remain separate at all refinement levels. This configuration tests the effect of grid resolution on the RT simulation. We adopt $\dot{N}_{\gamma, \text{ref}} = 8 \cdot 10^{53}$ photons \cdot s $^{-1}$, each point source is monochromatic with $E_\nu = 13.6$ eV, and the temperature is kept constant throughout the simulation. For simplicity, we consider a H only gas.

Figure 5 shows the maps of x_{HI} created in simulations

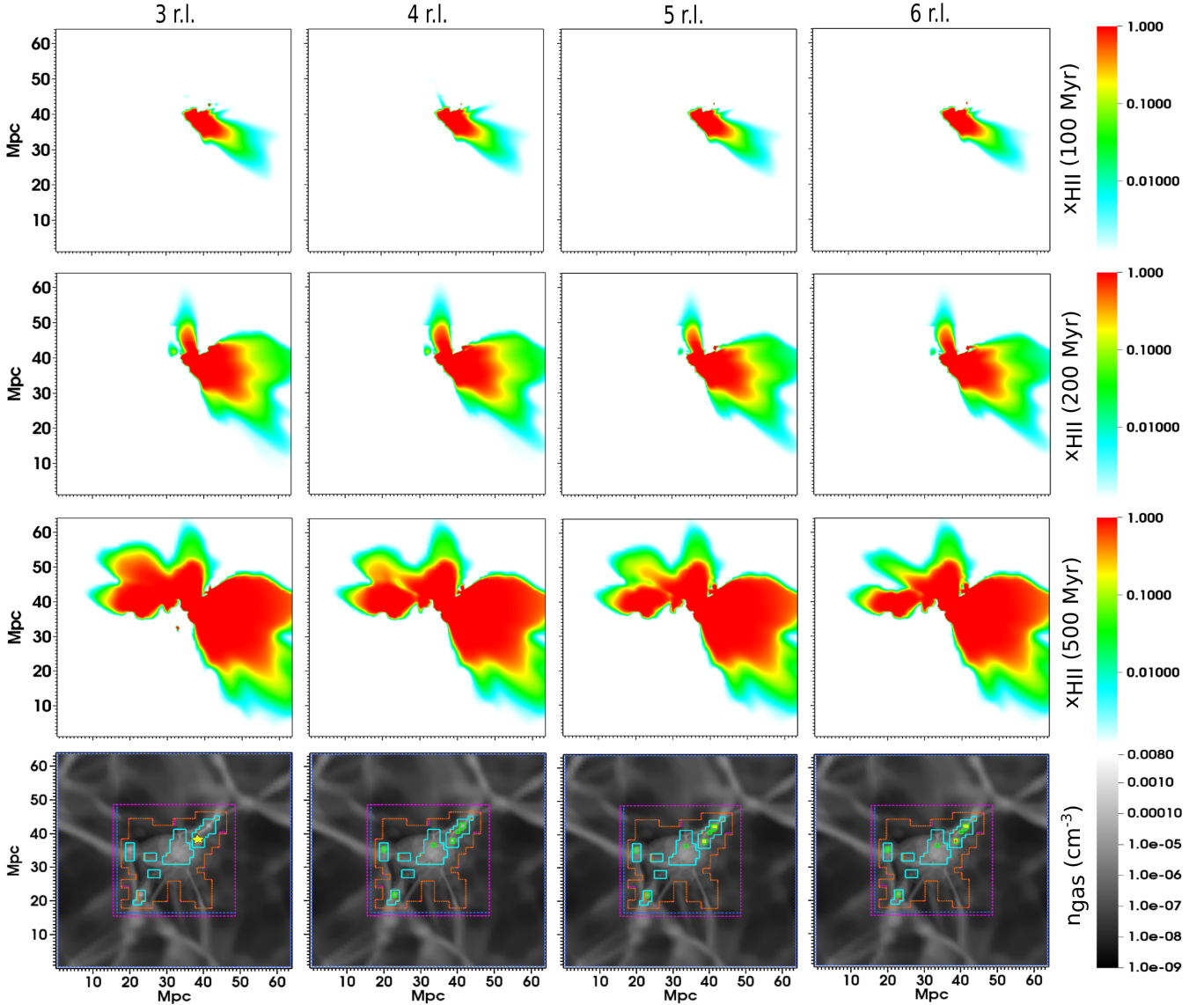


Figure 5. Maps cut through the simulation volume for Test 2a. **Top:** Maps of x_{HII} at time $t = 100$ Myr. **Second from top:** Maps of x_{HII} at time $t = 200$ Myr. **Third from top:** Maps of x_{HII} at time $t = 500$ Myr. **Bottom:** Maps of n_{gas} , the dotted lines represent the extent of the different refinement levels associated with n_{gas} (the base grid is not seen here): magenta (1st r.l.), orange (2nd r.l.), cyan (3rd r.l.), green (4th r.l.), yellow (5th r.l.) and red (6th r.l.). From left to right, the columns refer to simulations run with three, four, five and six refinement levels (see text for more details).

with different refinement levels at times $t = 100$, 200 and 500 Myr. In the bottom panels we also show the gas number density field (n_{gas}). Dotted lines represent the extent of the different refinement levels (see the caption for more details). For reference, we also show the location of the most luminous point source. Note that, because the RT is done in post-processing, the gas configuration does not change during the ionisation evolution.

At $t = 100$ Myr the x_{HII} maps are very similar. Differences become more visible at $t = 200$ Myr, where separate bubbles can be seen on the right side of the box as we go to higher refinement levels. The largest differences are present at the final time $t = 500$ Myr. From a comparison between the ionisation and gas number density maps, we can observe no direct correlation between positions of the refinement lev-

els and differences in the ionisation pattern, as the sources are able to maintain their surrounding regions fully ionised against the progressively steeper changes in density introduced by AMR. On the other hand, the extent of the fully and partially ionised H II regions shows obvious differences, as they get smaller and sharper with higher resolution. This is due to the larger changes in density and gas recombination rate (which increases by a factor of 3.5 between 6 and 3 r.l.) present in the more refined grids. As a result, the escape of ionising photons becomes more difficult, delaying the propagation of the ionisation fronts.

Finally, note that the presence of multiple point sources on different planes of the cube and resolved by different AMR layers, creates an intricate combination of three-dimensional RT effects in the final configuration of the over-

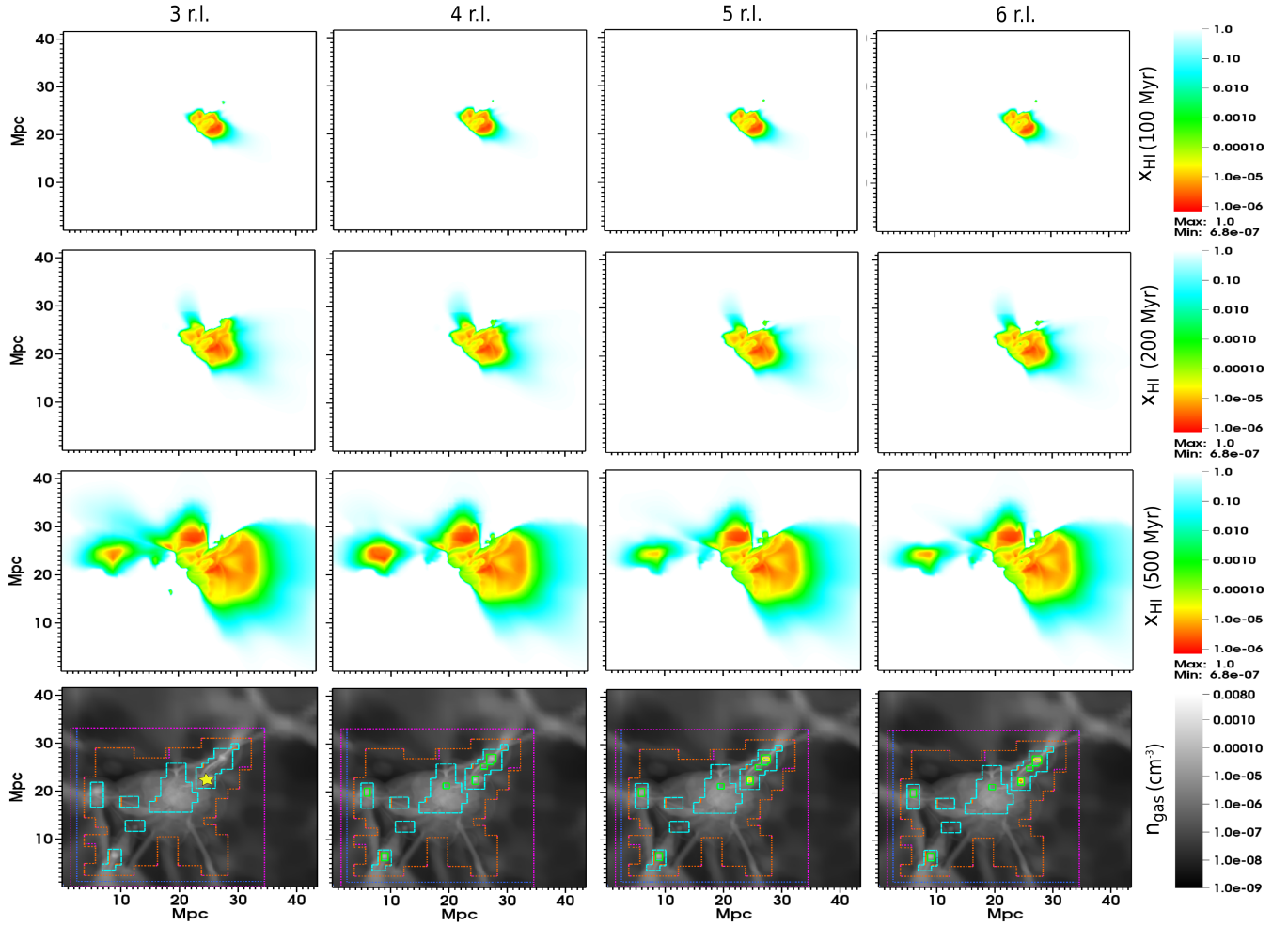


Figure 6. Maps cut through the simulation volume for Test 2a. **Top:** Maps of x_{HI} at time $t = 100$ Myr. **Second from top:** Maps of x_{HI} at time $t = 200$ Myr. **Third from top:** Maps of x_{HI} at time $t = 500$ Myr. **Bottom:** Maps of n_{gas} , the dotted lines represent the extent of the different refinement levels associated with n_{gas} (the base grid is not seen here): magenta (1st r.l.), orange (2nd r.l.), cyan (3rd r.l.), green (4th r.l.), yellow (5th r.l.) and red (6th r.l.). From left to right, the columns refer to simulations run with three, four, five and six refinement levels (see text for more details).

lapping H II fronts. This is more evident in the case with 5 and 6 refinements, in two distinct regions. The H II region on the left is in fact formed by a point source lying on a plane different from the one hosting the most luminous point source, and evolves differently with increasing number of refinement levels. This provides a final bubble distribution and overlap in space which is very sensitive to the number of adopted AMR refinements.

The effects of resolving more and more gas clumps by progressively increasing the AMR resolution are better appreciated by showing complementary $x_{\text{HI}} = 1 - x_{\text{HII}}$ maps. This is done in Figure 6, with the same panel organization of the previous figure. First note that the region shown here is closer to the image center; this is done to better zoom-in the spatial distribution of the neutral fraction at all times⁶. Also note that in these panels the colour palette indicates neutral gas fraction in logarithmic scale with a cyan-to-white tran-

sition corresponding to almost neutral gas, while orange-to-red areas mark almost ionised gas (i.e. $x_{\text{HI}} \lesssim 10^{-5}$). As commented above, the AMR refinements close to the sources can resolve more over-dense structures. While the radiation is sufficient to substantially ionise the entire area ($x_{\text{HII}} \gtrsim 0.9$) and to allow the escape of ionizing photons in far under-dense voids (see for example the one in the lower right side of the panels), many inner regions still show an intricate pattern of residual neutral gas: a large yellow area preserving a residual fraction $x_{\text{HI}} \sim 4 \times 10^{-4}$ surrounds the red spots and blends into green and cyan areas when the neutral fraction progressively increases up to $x_{\text{HI}} \sim 10^{-2}$ and $x_{\text{HI}} \sim 10^{-1}$, respectively. At megaparsec scales the structures in the various panels differ for only few, minor details, while the external contours show a clear reduction of the ionised gas (e.g. focus on the cyan area connecting the two central H II regions) from left (3 r.l.) to right (6 r.l.).

Also the volume averaged ionisation fraction depends on the refinement levels used, with $x_{\text{HII}} = 4.94, 4.74, 4.51$ and $4.39 \cdot 10^{-2}$ for the 3rd, 4th, 5th and 6th refinement lev-

⁶ As effect of the re-centering, the distance scale in Figure 6 does not correspond to the one in the previous figure.

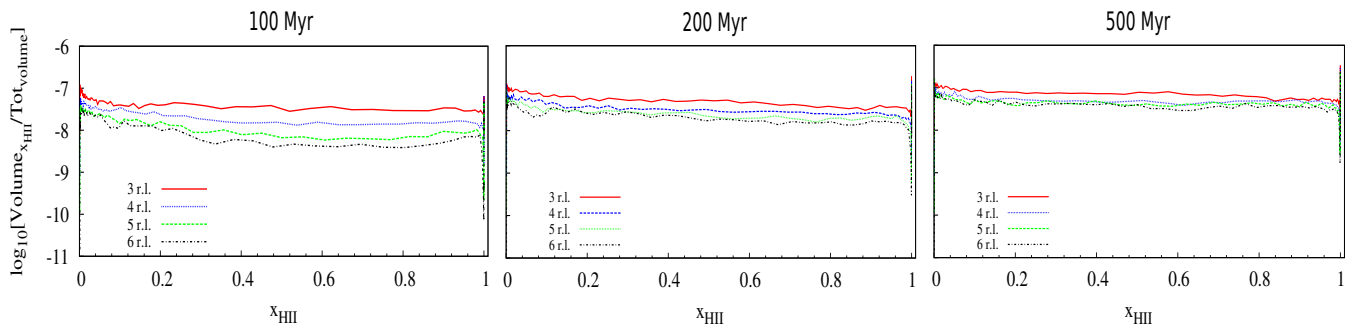


Figure 7. PDFs of x_{HII} in the simulation volume for Test 2a. Each panel refers to a different simulation time. **Left:** Simulation time $t = 100$ Myr. **Center:** Simulation time $t = 200$ Myr. **Right:** Final time $t = 500$ Myr. The lines refer to a test case with 3 (solid red), 4 (dotted blue), 5 (dashed green) and 6 (dot-dash black) refinement levels.

els at time $t = 500$ Myr, respectively, with a 12.5% difference between the test cases with 3 and 6 refinement levels. A better statistical description of the differences induced by an increase of AMR refinement levels is provided by the probability density functions (PDFs) of the ionised fraction x_{HII} . In the three panels of Figure 7 we show the ratio of the volume occupied by cells with a given x_{HII} over the total volume at $t = 100, 200$ and 500 Myr. Note that the PDFs reflect the trend of Figure 5, with the largest (smallest) volume occupied by ionised cells in the least (most) refined configuration. Also note that the evolution in time of ionised regions reduces the differences while maintaining the trends across refinement levels.

Figure 8 shows the x_{HII} profile for three randomly selected line of sights (LOSs) from the most luminous point source in the simulation. The trend observed in the LOSs reflects the above comments. In all the panels, the ionisation front (I-front⁷) systematically recedes with a larger number of refinement levels, due to the increased gradient in n_{gas} and the higher gas recombination rate. We note that, although the above trend is observed in the vast majority of the LOSs, there are exceptions due to the large variety of gas properties. We find that the distance from the source location to the I-front position for the test cases with 6 refinement levels is smaller than that with 3 refinement levels by 6.8, 4.39 and 13.6% from the top to the bottom panel.

We then conclude that the photo-ionisation algorithm of CRASH-AMR is highly sensitive to the changes in the gas number density resolved by more refinement levels; this is reflected in the variations observed in the ionisation structures and I-fronts. CRASH-AMR will provide a more precise and realistic representation of how the ionised bubbles form and grow around the high density regions in which star formation occurs, as well as a better estimate of the escape of ionising photons through the IGM when local-scale reionisation simulations will be performed with this technique.

⁷ We define the I-front as the point at which the x_{HII} drops below 0.5.

5.2.2 Test 2b: multiple sources set close to each other - constant T

Differently from Test 2a, here we place the twenty sources so that they are close enough at the highest resolution to be gathered at lower refinement levels. This results in 12, 6 and 2 sources at the 5th, 4th and 3rd refinement level, respectively. Here we assume $\dot{N}_{\gamma_{\text{ref}}} = 5 \cdot 10^{53} \text{ photons} \cdot \text{s}^{-1}$, while the rest of the set-up is the same as in Test 2a.

Since in this test multiple sources are represented at the coarser levels as a single one of higher luminosity, at early times we expect to see the growth of only one ionised region at the coarser level, whereas at higher levels the ionised regions remain distinct from each other. To capture these features, in Figure 9 we show maps of x_{HII} at $t = 10$ Myr. While at low resolution we find a single ionised region, with x_{HII} in the range $\sim 0.8-1$, at higher resolutions a much smaller region has such high ionisation fraction. This translates into a volume averaged x_{HII} fraction of 1.38, 1.43, 1.38 and 1.37 $\cdot 10^{-3}$ for the 3rd, 4th, 5th and 6th refinement level, respectively, with a 0.7% difference between the 3rd and 6th levels. At 500 Myr, the differences in the x_{HII} fraction averages are 1.1%. Panel insets show the distribution of neutral gas in the zoomed regions. Note that here the colour coding shows areas with $x_{\text{HI}} < 10^{-3}$ in red, while white areas represent almost neutral regions. Also note how the increased refinement progressively confines the ionised areas at the center.

We have compared the PDFs of x_{HII} between the different test cases and find a trend similar to the one of Test 2a, i.e., the configuration with 6 r.l. shows the least volume occupied by fully ionised cells.

Our comparison of the LOSs between the different test cases also shows a trend similar to Test 2a, i.e., the largest differences are observed in the partially ionised gas, with the extent of the ionised region receding with increasing resolution.

5.2.3 Test 2c: multiple sources set far apart - T calculated self-consistently

The sources here are located as in Test 2a, but now they have a black-body spectrum at temperature $T_{\text{BB}} = 10^5$ K and the gas temperature is calculated self-consistently with the progress of ionisation during the simulation.

Figures 10 and 11 show the maps of x_{HII} and T created

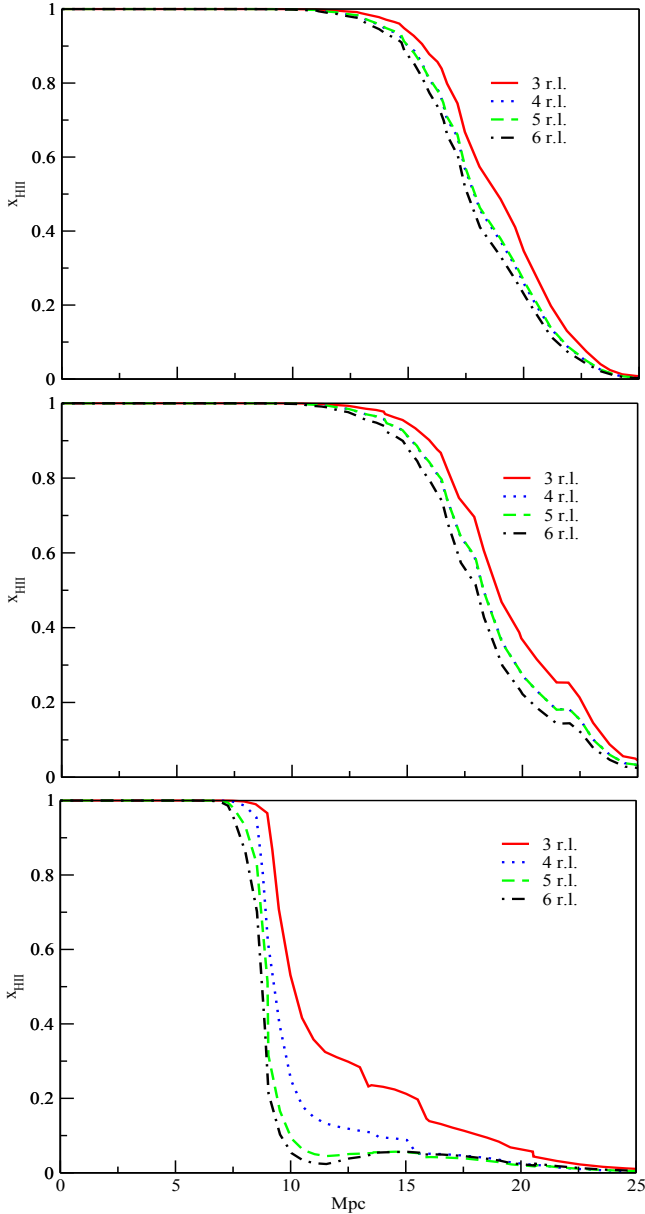


Figure 8. x_{HII} profiles at time $t = 500$ Myr for three LOSs in random directions from the most luminous point source in Test 2a. The lines refer to a test case with three (solid red), four (dotted blue), five (dashed green) and six (dot-dash black) refinement levels.

in simulations with different refinement levels at a time $t = 100, 200$ Myr and 500 Myr. We show the same slices of Figure 5, i.e., containing the most luminous point source. Hence, the density field corresponds to the bottom slice in Figure 5.

Similarly to Test 2a, we find that at a time $t = 100$ Myr the maps look very similar. At $t = 200$ Myr, the ionised regions start to show some differences, noticeably in the separate bubble that is formed in all the slices. The same differences can be found in the maps of T as well. At $t = 500$ Myr, the most obvious differences are found in the H II region on the left that is formed by a point source lying on a plane different from the one shown in the Figure. The correspond-

ing T maps have a similar behaviour. Finally note that in each panel we provided a zoom-in view of the central region showing the residual distribution of neutral gas. As discussed above, many areas with residual neutral gas are found in the inner region, with values in the range $10^{-5} < x_{\text{HI}} < 10^{-4}$.

The differences are reflected also in the volume averaged ionisation fraction, which is $x_{\text{HII}} = 2.54, 2.49, 2.39$ and $2.34 \cdot 10^{-2}$ for 3, 4, 5 and 6 refinement levels at time $t = 500$ Myr, respectively, with a 8.5% difference between the 3rd and 6th levels. The corresponding temperatures are 2264, 2221, 2189 and 2159 K, with a 4.8% difference between the 3rd and 6th levels. Comparing Figures 5 and 10 we find that the size of the H II region is smaller in the latter case, as a result of the spectral distribution of the ionising sources and the self-consistent calculation of the gas temperature: the introduction of a spectral distribution creates a number of high-energy photon packets which can easily diffuse beyond the ionisation fronts creating wider regions at low ionisation, while the various cooling processes which are being taken into account have a substantial feedback on the gas recombination rates creating smoother ionisation gradients. We do expect these differences to be even more marked in the presence of Helium.

In Figure 12 we show x_{HII} along the three LOSs of Figure 10. The corresponding T is shown in Figure 13. A trend similar to the one of Test 2a is observed, i.e., the extent of the fully ionised region becomes smaller with increasing refinement levels. The T profiles exhibit the same behaviour, with a good agreement within the fully ionised region and a T decreasing with increasing refinement levels. As already mentioned in the description of the test setups, the base grid in use is refined only in the central region, close to the sources. As a consequence, the resolution at large distances is the same in all cases, and the discrepancies are due to the differences experienced by the various photon packets during their propagation through the central region rather than to local density variations.

6 CONCLUSION

In this paper we have introduced **CRASH-AMR**, a new version of the cosmological radiative transfer code **CRASH**, enabled to run RT simulations on AMR grids. After an exhaustive discussion of the code, we have shown the results of many tests both with the simplified set-up prescribed in the Radiative Transfer Code Comparison Project and a realistic hydrodynamic simulation with AMR refinement. All the tests show good agreement with the latest release of **CRASH**, confirming the correct inclusion of a more accurate and alternative geometry representation of the gas distribution in the cosmological domain in which the RT simulation is performed. The small discrepancies found are due either to the presence of a grid at higher resolution in the refined levels or to averaging operations.

The application to a realistic density field shows differences in the pattern of the ionised regions because of the more accurate treatment of the gas optical depth and cooling function. Consequently, at higher resolution the gas ionisation fractions and temperature are calculated with greater accuracy, allowing a better modeling of the growth of ionised bubbles, as well as of the escape of ionising radiation from

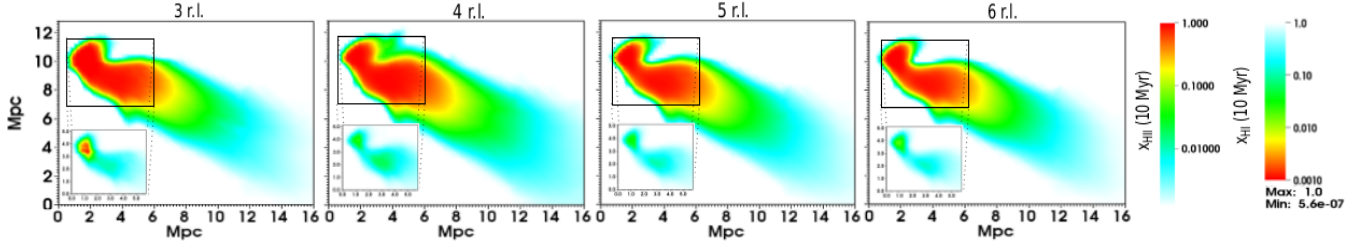


Figure 9. Maps of x_{HII} for Test 2b at time $t = 10$ Myr. From left to right, the columns refer to simulations run with three, four, five and six refinement levels (see text for more details). Panel insets show the neutral fraction of the gas in the zoom-in areas; their colour coding refers to the second palette on the right. Min and max values, when not represented by the palette, are written as reference.

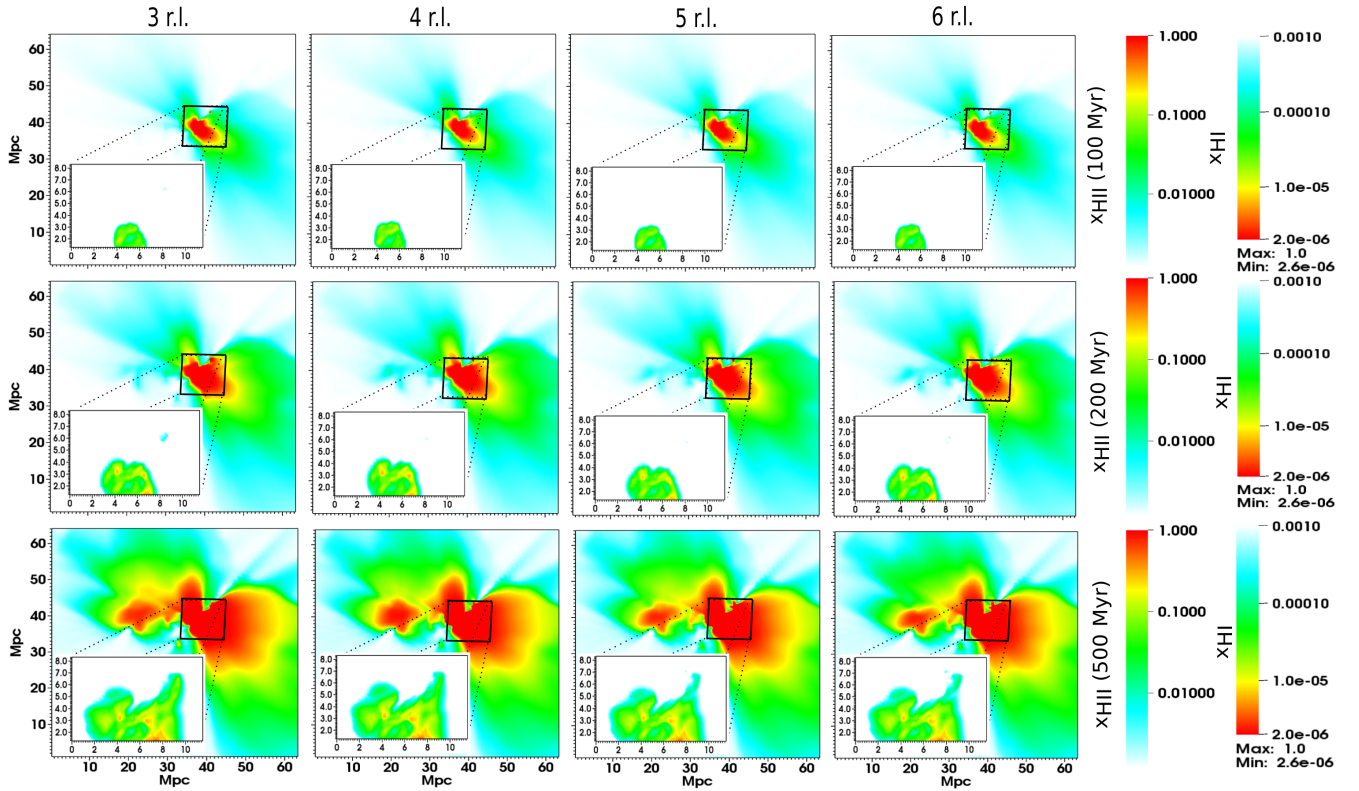


Figure 10. Maps cut through the simulation volume for Test 2c. **Top:** Maps of x_{HII} at time $t = 100$ Myr. **Middle:** Maps of x_{HII} at time $t = 200$ Myr. **Bottom:** Maps of x_{HII} at time $t = 500$ Myr. From left to right, the columns refer to simulations run with three, four, five and six refinement levels (see text for more details). Panel insets show the neutral fraction of the gas in the zoom-in areas; their colour codings refer to the second palette on the right. Min and max values, when not represented by the palette, are written as reference.

high-density regions in which star formation is typically embedded. In general, **CRASH-AMR** will provide an invaluable improvement compared to the previous algorithm whenever a better resolution of the radiation-matter interaction in some specific regions is needed.

As final consideration, we note that **CRASH-AMR** is able to perform RT simulations in high density regions with the resolution increased by a factor of 64 with respect to the base 64^3 grid resolution without experiencing serious memory limitations. Such a high resolution would be unmanageable from the storage point of view in a single CPU core without this new version of the code. We also find that by running a specific configuration on smarter AMR grids

rather than on uniform 512^3 grids, a $\sim 60\%$ reduction in run time is obtained. Hence, **CRASH-AMR** provides an advantage both in terms of memory consumption and run time performance when compared to the standard version of **CRASH**, especially in the future releases where **CRASH-AMR** will take advantage of distributed memory parallelism using MPI.

ACKNOWLEDGEMENTS

N.H thanks the **CHOMBO** and **FLASH** development team for their help in using **CHOMBO** with a Fortran code. N.H also thanks Koki Kakiuchi and Prof. Michael Bader for their use-

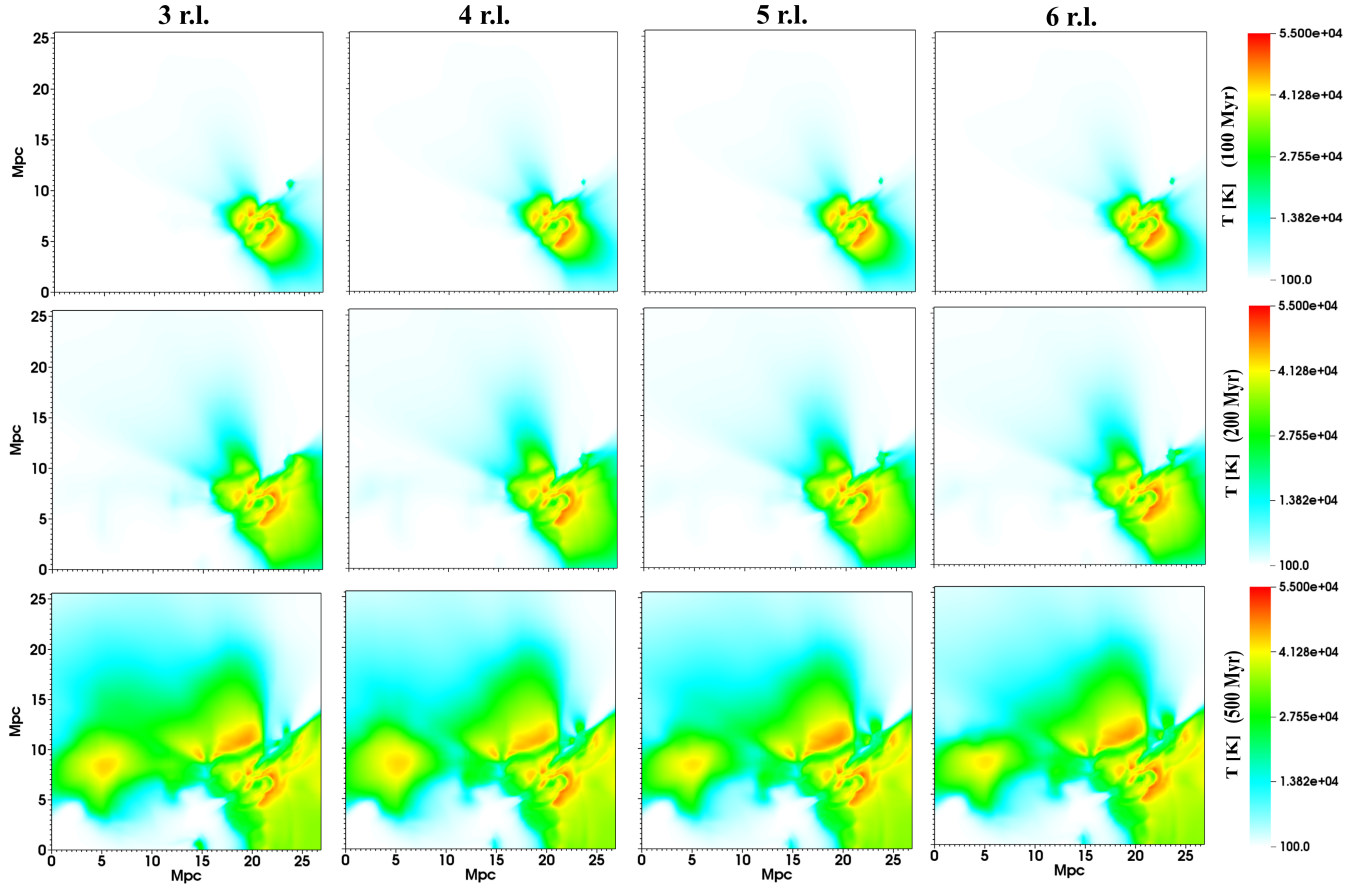


Figure 11. Maps cut through the simulation volume for Test 2c. **Top:** Maps of T at time $t = 100$ Myr. **Middle:** Maps of T at time $t = 200$ Myr. **Bottom:** Maps of T at time $t = 500$ Myr. From left to right, the columns refer to simulations run with three, four, five and six refinement levels (see text for more details).

ful suggestions. LG acknowledges the support of the DFG Priority Program 1573 and the support of the European Research Council under the European Union (FP/2007-2013) / ERC Grant Agreement n. 306476.

REFERENCES

- Abel T., Wandelt B. D., 2002, *MNRAS*, **330**, 53
- Adams M., et al., 2011, CHOMBO Software Package for AMR Applications - Design Document, *Lawrence Berkeley National Laboratory Technical Report LBNL-6616E*
- Ahn K., et al., 2012, *APJL*, **756**, L16
- Alvarez M. A., Bromm V., Shapiro P. R., 2006, *ApJ*, **639**, 621
- Angulo R. E., et al., 2012, *MNRAS*, **426**, 2046
- Barkana R., Loeb A., 2007, *Reports on Progress in Physics*, **70**, 627
- Berger M., Colella P., 1989, *Journal of Computational Physics*, **82**, 64
- Berger J. M., Olinger J., 1984, *Journal of Computational Physics*, **53**, 484
- Boylan-Kolchin M., Springel V., White S. D. M., Jenkins A., Lemson G., 2009, *MNRAS*, **398**, 1150
- Bryan G. L., et al., 2014, *ApJS*, **211**, 19
- Cantalupo S., Porciani C., 2011, *MNRAS*, **411**, 1678
- Ciardi B., Ferrara A., 2005, *SSR*, **116**, 625
- Ciardi B., Ferrara A., Marri S., Raimondo G., 2001, *MNRAS*, **324**, 381
- Ciardi B., Stoehr F., White S. D. M., 2003, *MNRAS*, **343**, 1101
- Compostella M., Cantalupo S., Porciani C., 2013, *MNRAS*, **435**, 3169
- Dai W. W., 2010, in *IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW)*, pp 1–8, doi:10.1109/IPDPSW.2010.5470758
- Davé R., 2005, in Colless M., Staveley-Smith L., Stathakis R. A., eds, *IAU Symposium Vol. 216, Maps of the Cosmos*. p. 251
- Dubey A., et al., 2014, *Journal of Parallel and Distributed Computing*, **74**, 3217
- Fan X., et al., 2006, *AJ*, **132**, 117
- Frenk C. S., et al., 1999, *APJ*, **525**, 554
- Fryxell B., et al., 2000, *ApJS*, **131**, 273
- Gnedin N. Y., 2014, *ApJ*, **793**, 29
- Gnedin N. Y., Abel T., 2001, *New Astronomy*, **6**, 437
- Gnedin N. Y., Kaurov A. A., 2014, *ApJ*, **793**, 30
- Gnedin N. Y., Ostriker J. P., 1997, *ApJ*, **486**, 581
- Gnedin N. Y., Tassis K., Kravtsov A. V., 2009, *APJ*, **697**, 55
- Górski K. M., et al., 2005, *ApJ*, **622**, 759
- Graziani L., Maselli A., Ciardi B., 2013, *MNRAS*, **431**, 722
- Graziani L., et al., 2015, *MNRAS*, **449**, 3137
- Harnois-Déraps J., et al., 2013, *MNRAS*, **436**, 540
- Iliev I. T., et al., 2006a, *MNRAS*, **369**, 1625
- Iliev I. T., et al., 2006b, *MNRAS*, **369**, 1625
- Iliev I. T., Mellema G., Shapiro P. R., Pen U.-L., 2007, *MNRAS*, **376**, 1055

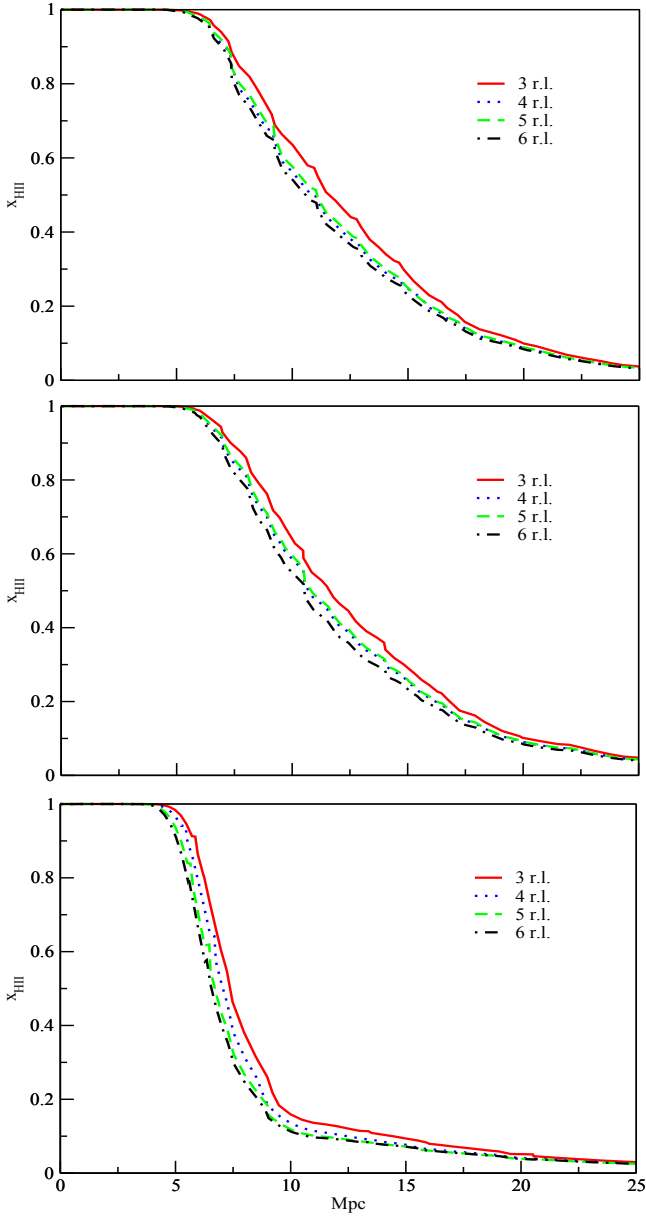


Figure 12. x_{HII} profiles at time $t = 500$ Myr for three LOSs in random directions from the most luminous point source in Test 2c. The lines refer to a test case with three (solid red), four (dotted blue), five (dashed green) and six (dot-dash black) refinement levels.

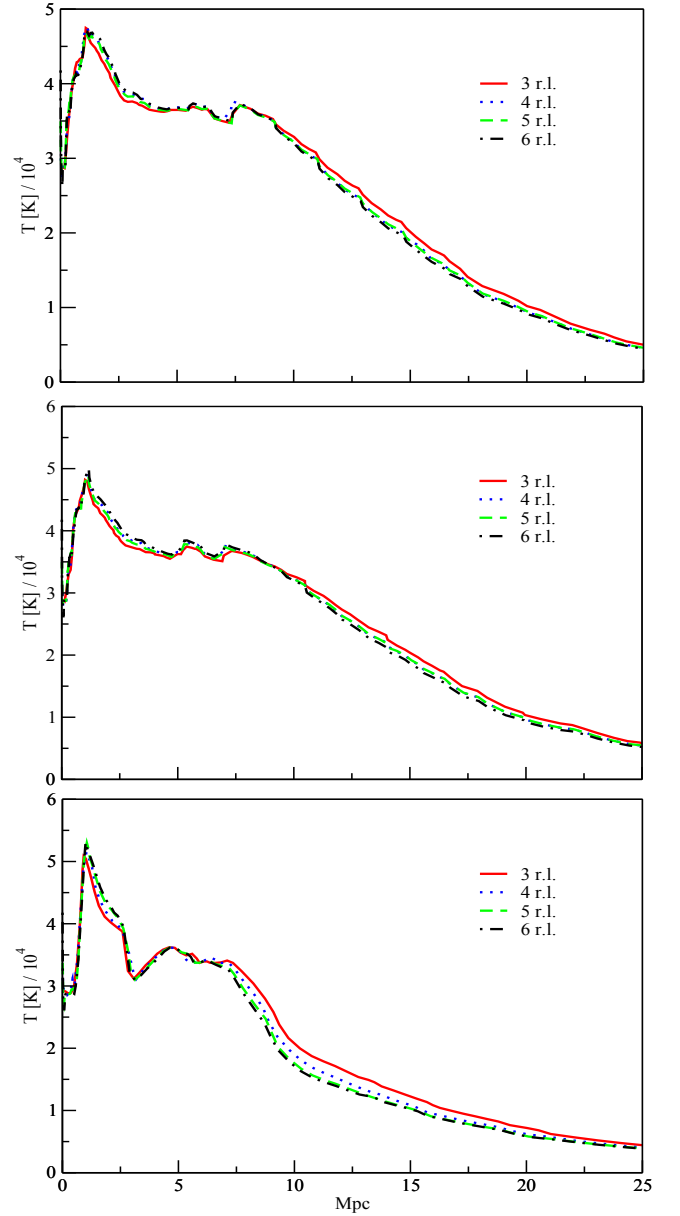


Figure 13. T profiles at time $t = 500$ Myr for three LOSs in random directions from the most luminous point source in Test 2c. The lines refer to a test case with three (solid red), four (dotted blue), five (dashed green) and six (dot-dash black) refinement levels.

376, 534

Iliev I. T., et al., 2009, *MNRAS*, **400**, 1283

Iliev I. T., et al., 2014, *MNRAS*, **439**, 725

Khokhlov A., 1998, *Journal of Computational Physics*, **143**, 519

Kim J., et al., 2011, *Journal of Korean Astronomical Society*, **44**, 217

Kravtsov A. V., Klypin A. A., Khokhlov A. M., 1997, *ApJS*, **111**, 73

Lee D., Deane A. E., 2009, *Journal of Computational Physics*, **228**, 952

Levermore C. D., 1984, *JQSRT*, **31**, 149

Madau P., Meiksin A., 1994, *APJL*, **433**, L53

Maselli A., Ferrara A., 2005, *MNRAS*, **364**, 1429

Maselli A., et al., 2003, *MNRAS*, **345**, 379

Maselli A., Ciardi B., Kanekar A., 2009, *MNRAS*, **393**, 171

Mavriplis D. J., 1997, *Annual Review of Fluid Mechanics*, **29**, 473

Meiksin A. A., 2009, *Reviews of Modern Physics*, **81**, 1405

Mignone A., et al., 2012, *ApJS*, **198**, 7

Miniati F., 2014, *APJ*, **782**, 21

Miniati F., Colella P., 2007, *Journal of Computational Physics*, **227**, 400

Miniati F., Martin D. F., 2011, *APJS*, **195**, 5

Mo H., van den Bosch F. C., White S., 2010, *Galaxy Formation and Evolution*. Cambridge University Press

Paardekooper J.-P., Icke V., Ritzerveld J., 2008, in O'Shea B. W., Heger A., eds, *American Institute of Physics Conference Series* Vol. 990, First Stars III. pp 453–455, doi:10.1063/1.2905662

Partl A. M., Maselli A., Ciardi B., Ferrara A., Müller V., 2011,

MNRAS, 414, 428

Pawlik A. H., Schaye J., Dalla Vecchia C., 2015, MNRAS, 451, 1586

Pierleoni M., Maselli A., Ciardi B., 2009, MNRAS, 393, 872

Plewa T., Timur L., Gregory V. W., 2003, in Proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods.

Razoumov A. O., Cardall C. Y., 2005, MNRAS, 362, 1413

Rosdahl J., Blaizot J., Aubert D., Stranex T., Teyssier R., 2013, MNRAS, 436, 2188

Schaye J., et al., 2014, preprint, (arXiv:1407.7040)

Semelin B., Combes F., Baek S., 2007, A&A, 474, 365

Shaw L. D., et al., 2006, ApJ, 646, 815

Springel V., 2005, MNRAS, 364, 1105

Springel V., 2011, in Alves J., Elmegreen B. G., Girart J. M., Trimble V., eds, IAU Symposium Vol. 270, Computational Star Formation. pp 203–206, doi:10.1017/S1743921311000378

Teyssier R., 2002, A&A, 385, 337

Teyssier R., Fromang S., Dormy E., 2006, Journal of Computational Physics, 218, 44

Trac H., Cen R., 2008, in O'Shea B. W., Heger A., eds, American Institute of Physics Conference Series Vol. 990, First Stars III. pp 445–449, doi:10.1063/1.2905660

Vogelsberger M., et al., 2014, NATURE, 509, 177

Wise J. H., Abel T., 2011, MNRAS, 414, 3458

Wu K. K. S., Lahav O., Rees M. J., 1999, Nature, 397, 225

Young P. D., et al., 1991, Journal of Computational Physics, 92, 1

Zahn O., et al., 2007, APJ, 654, 12

APPENDIX A: DEPENDENCE ON GRID RESOLUTION

The difference in results observed in Section 5.1 is only due to the grid resolution and not to the new CRASH-AMR implementation. While some discrepancies are expected when the tests are set-up with different base grid resolutions, this should not be the case when the resolution at the finest refinement level is the same. We demonstrate this by setting up some test cases similar to Test 1, where the resolution at the base grid is different, but that at the finest AMR level is the same, i.e. 512^3 . We thus expect the RT simulations to give the same results. Note that here we have completely refined the grid, although this is not usually done with AMR codes. We set up test cases with the following grid properties:

- (a) base grid resolution 64^3 , three levels of refinement;
- (b) base grid resolution 128^3 , two levels of refinement;
- (c) base grid resolution 256^3 , one refinement level;
- (d) base grid resolution 512^3 , no refinement.

Figure A1 shows the spherical averages of x_{HII} , x_{HI} , T , x_{HeII} and x_{HeIII} from the above simulations. We have calculated the spherical average at the highest resolution for all cases, and, as expected, it is exactly the same for the different base grid resolutions.

APPENDIX B: CODE PERFORMANCE

In this section we investigate the run time performance of CRASH-AMR while ensuring correctness in results. In Section 4 we mentioned that there was no run time overhead from the coupling of CHOMBO to CRASH, but only the additional time to search for the new cell in the multiple refinement levels that

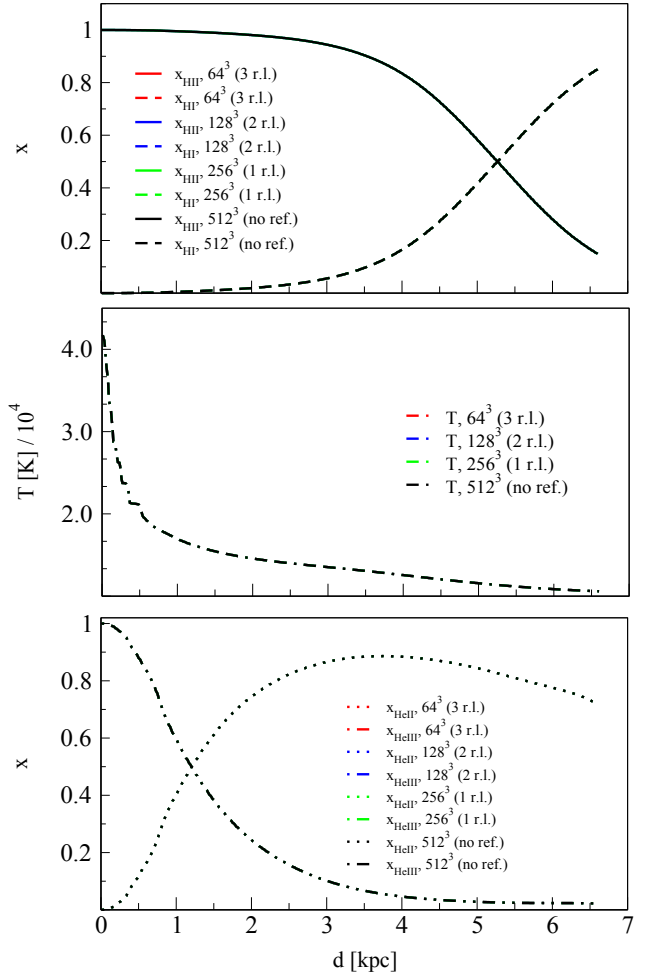


Figure A1. Spherically-averaged profiles at time $t = 500$ Myr for Test 1b. The lines refer to CRASH-AMR with different base grid resolutions, 64^3 (four AMR levels - red), 128^3 (three AMR levels - blue), 256^3 (two AMR levels - green) and 512^3 (no refinement - black). **Top:** Profiles of x_{HII} (solid lines) and x_{HI} (dashed lines). **Middle:** Profile of T (dash-dot-dash lines). **Bottom:** Profiles of x_{HeII} (dotted lines) and x_{HeIII} (dot-dash lines).

the ray crosses. Even though this search adds to the total run time in the RT simulations done using AMR grids, its effect is small when compared to running the RT simulation on a uniform high resolution grid. To prove that this is indeed the case, we ran some tests comparing the run times between grids at uniform resolution and AMR grids from the CHARM simulation. We then compare the two test cases to ensure that CRASH-AMR is able to provide accurate results.

B1 Set up with a single point source

The set-up is similar to Test 2a, with the ionised region from a single point source expanding into a realistic density field. The source is placed in the highest density peak, with $\dot{N}_\gamma = 5 \cdot 10^{54} \text{ photons} \cdot \text{s}^{-1}$, a monochromatic spectrum of $E_\nu = 13.6 \text{ eV}$ and emits $N_p = 2 \cdot 10^8$ photon packets. The gas is assumed to contain only H with an initial ionisation fraction x_{HII} set to $1.2 \cdot 10^{-3}$. The gas temperature T is initially set to 100 K and is kept constant throughout the simulation. The simulation time is $t_{\text{sim}} = 500 \text{ Myr}$, starting at redshift

Table B1. Run times (in hours) and the corresponding volume averaged ionisation fraction x_{HII} at time $t = 500$ Myr for CRASH-AMR simulations run on a configuration similar to that in Test 2a.

Test case	Grid resolution	Run time (hours)	x_{HII}
<i>U128</i>	Uniform grid (UG), 128^3	1.60	0.181
<i>R128</i>	Refined grid (RG), 128^3	1.58	0.180
<i>U256</i>	UG, 256^3	3.55	0.185
<i>R256</i>	RG, 256^3	2.52	0.184
<i>U512</i>	UG, 512^3	8.32	0.183
<i>R512</i>	RG, 512^3	3.40	0.181

$z = 0.1$. We output the results at intermediate times $t = 10, 20, 100, 200$ and 500 Myr.

To compare the run times we have created uniform resolution grids from the AMR refined grids used in Test 2 by interpolating the coarse data onto the finer levels. We use uniform grids of resolution 128^3 (*U128*), 256^3 (*U256*) and 512^3 (*U512*), and compare the run times to those of grids with base resolution of 64^3 and 1 (*R128*), 2 (*R256*) and 3 (*R512*) refinement levels, respectively. We would like to point out that there are other factors that also impact the run time of a CRASH simulation, for example the number of point sources, presence of He with T evolution and number of photon packets. However, we do not consider them here and focus instead on the performance of the code for a simple test case. Table B1 shows the resulting run times together with the corresponding volume averaged ionisation fraction x_{HII} at time $t = 500$ Myr.

We find that the difference in run time between *U128* and *R128* is only 1.25%, but this increases as we move to higher resolutions, and reaches 59% for *U512* and *R512*. This large gain in computational speed does not come at the expense of correctness of results. In fact, the difference between the uniform and refined grids in terms of volume average ionisation fraction is only 0.5, 0.5 and 1.1%. Additionally, two random LOSs are compared in Figure B1 for the three different resolutions. We find that the extent of the fully ionised H II region for the uniform and AMR grid cases is the same at all resolutions, while the partially ionised regions show some discrepancy. The I-front in the refined grids is in fact smaller than that in the uniform grids, with a maximum difference of 2%.

B2 Set up with multiple point sources

Here we use again a setting similar to that of Test 2a, and compared the results between a 512^3 uniform grid and a grid with a base resolution of 64^3 with 3 refinement levels. Table B2 shows the run times together with the corresponding x_{HII} at time $t = 500$ Myr.

Here again we find that the difference in run times and volume average ionisation fraction between the two cases is as high as 38% and 1.3%, respectively. A comparison between three random LOSs from the most luminous source (Fig. B2) confirms that the extent of the fully ionised regions is the same with and without uniform grids, while the maximum difference in the size of the I-front is 2%.

From the two test cases above it is clear that CRASH-AMR provides results consistent with those from CRASH3 but with much shorter run times. We have been able to represent the

regions of interest at a resolution increased by a factor of 64 and not lose any accuracy in the results⁸. Our interface with CHOMBO and the ray-tracing algorithm to search for the new cell across different refinement levels during ray-tracing is highly efficient and does not have a negative impact on the run time.

We would, however, like to point out that in both the test cases above the decrease in run times between the uniform resolution and AMR grids is not proportional to the percentage decrease in the number of cells in the AMR grids. For example, in test case *U512* the number of cells in the grid is 134,217,728 while that in *R512* is 1,816,576 which is a 98% decrease in the number of cells. However, the corresponding decrease in run times is 60%. This is due to the fact that the run time now includes:

- the time taken for the initial setup, which includes setting up the AMR hierarchy in CRASH-AMR. This is done once at the beginning of the simulation;
- the time taken to find the right box that the ray is in, every time it enters a new cell. This step includes a number of other checks, which have been discussed in section 4.

Also, as of now, we have implemented a simple search routine in `b` to loop through an unsorted neighbor and parent lists until the correct box is found. As the PBAMR scheme allows for multiple neighbor and parent boxes, this search can be expensive if done frequently. We are looking at optimising this search by sorting the neighbor and parent lists in the 3D space, by using the Morton space-filling curve implemented in CHOMBO, before using them to search for the new box.

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.

⁸ Note here that the accuracy of the results is very problem dependent.

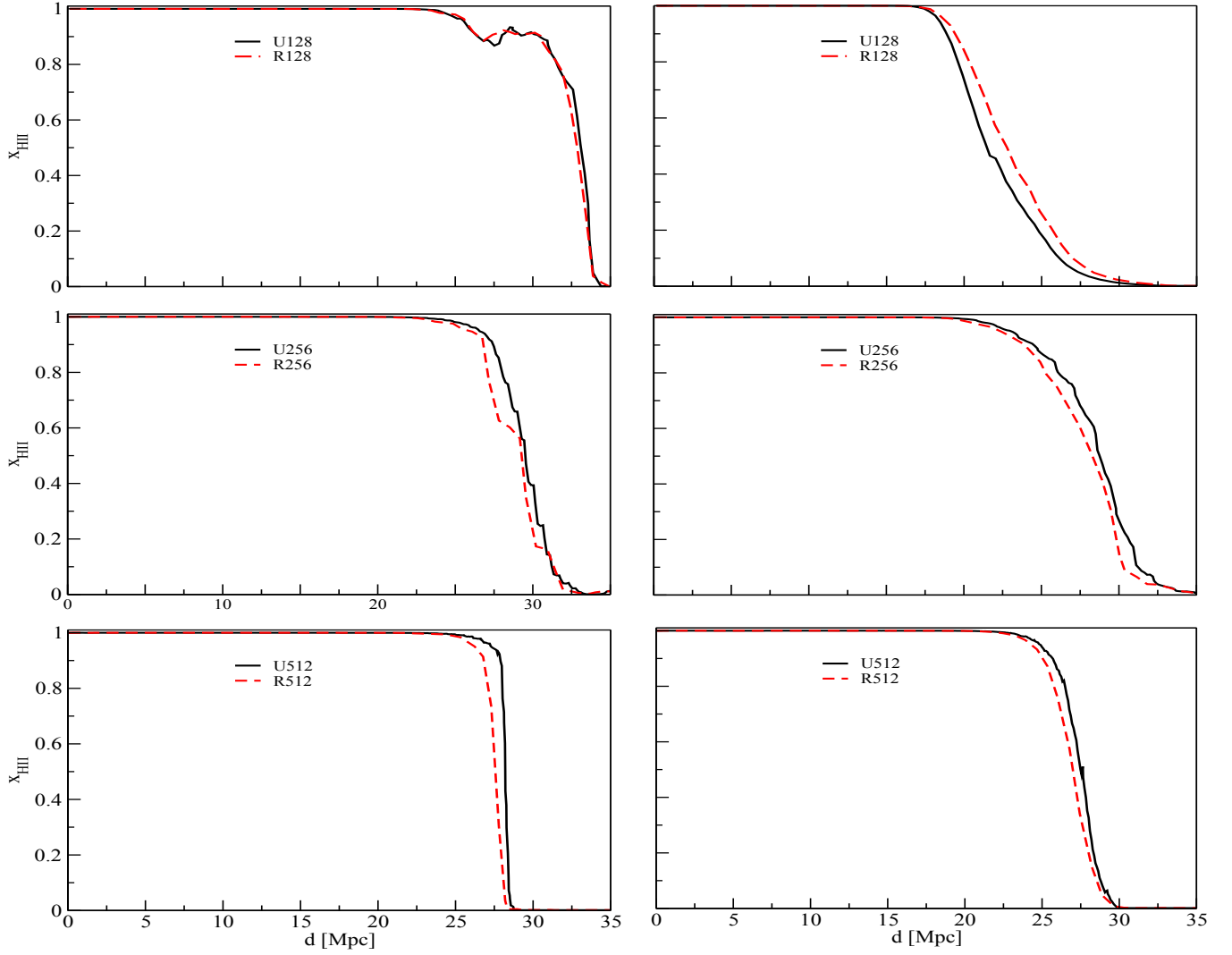


Figure B1. LOSs in two random directions from a point source for Test 2a at time $t = 500$ Myr. The lines refer to CRASH-AMR with a uniform grid (solid black) and with refinement levels (dashed red). **Top:** *U128* and *R128*. **Middle:** *U256* and *R256*. **Bottom:** *U512* and *R512*.

Table B2. Run times (in hours) and the corresponding x_{HII} at time $t = 500$ Myr for CRASH-AMR simulations run on a configuration similar to that in Test 2a.

Test case	Grid resolution	Run time (hours)	x_{HII}
<i>U512</i>	UG, 512^3	11.4	0.0501
<i>R512</i>	RG, 512^3	6.97	0.0494

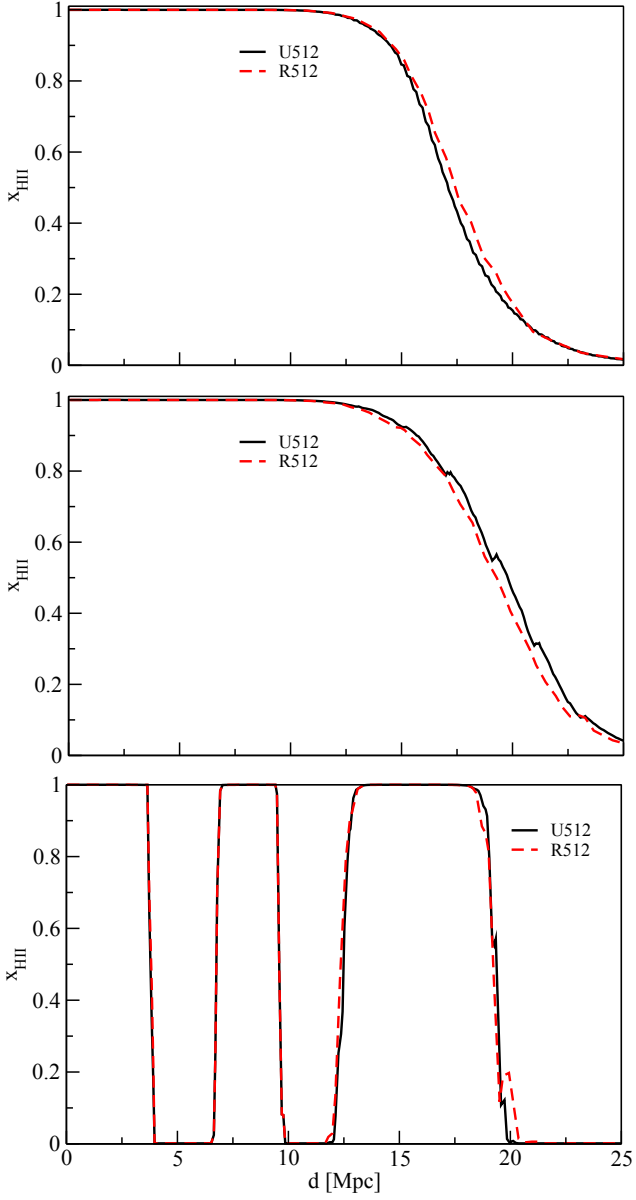


Figure B2. LOSs in random directions from a point source for Test 2a at time $t = 500$ Myr. The lines refer to **CRASH-AMR** with a uniform grid (solid black) and with 3 r.l. (dashed red).